

VŠB – Technická univerzita Ostrava

Fakulta strojní

Katedra mechaniky



Analýza a syntéza mechanismů

Analysis and Synthesis of Mechanisms

Disertační práce

Studijní program: P2301 Strojní inženýrství

Studijní obor: 3901V003 Aplikovaná mechanika

Školitel: Doc. Ing. Jiří PODEŠVA, Ph.D.

Doktorand: Ing. Ondřej FRANTIŠEK

Ostrava, prosinec 2011



## **Anotace**

Disertační práce se zabývá analýzou a syntézou kinematických vlastností komplikovaných, prostorových mechanismů. V první části textu byla vypracována metodika analýzy mechanismů metodou transformačních rovnic. Dále je navržena metoda řešení úlohy polohy pro analýzu mechanismů s cílem dosažení co nejvyšší efektivity a tímto snížit výpočtové časy. K syntéze mechanismů jsou využity metody matematické teorie optimalizace, jako metoda simplexů, metoda BFGS, metoda penalizace atd. Pozornost je věnována i komplikacím, které se během použití optimalizace při syntéze mechanismů vyskytují, a jsou navržena také jejich možná řešení. V rámci práce jsou řešeny tři problémy technické praxe, a to prohození mechanismus tkacího stroje a dva mechanismy řazení. Neopomenutelným přínosem práce jsou algoritmy a procedury popsány v práci a v přílohách práce.

## **Abstract**

Subject of the dissertation thesis are analysis and synthesis of kinematical characteristics of mechanisms. The methodology of mechanisms analysis by method of transformation matrices is presented in the first chapter of the work. The numerical method for solving mechanisms position is suggested with respect to achieve high efficiency and low computing times. Optimization methods are used for synthesis of mechanisms. These methods are the simplex method, the BFGS method and the method of penalties. Attention is also paid to complications occurring during optimization process and methods avoiding these ones are suggested. The thesis is concluded by three problems from practice. The problems describe pitching mechanism and two shifting mechanisms. Algorithms and procedures are presented in the work.





### *Poděkování*

*Chtěl bych poděkovat panu docentu Jiřímu Podešvovi za cenné připomínky, vřelý přístup a příjemně strávené chvíle nad bakalářskou, diplomovou a nyní doktorskou prací.*

## Obsah

Seznam algoritmů.....	11
Seznam použitého značení.....	13
1 Úvod.....	19
2 Rešerše a cíle práce.....	21
2.1 Výsledky rešerše.....	21
2.2 Cíle práce.....	22
3 Analýza kinematických vlastností prostorových mechanismů.....	24
3.1 Základní pojmy teorie mechanismů.....	24
3.2 Vlastnosti mechanismů.....	27
3.2.1 Prostorové vazby a jejich schematické značky.....	27
3.2.2 Kinematické schéma mechanismu.....	28
3.2.3 Počet stupňů volnosti mechanismu.....	29
3.2.4 Kinematické řetězce a smyčky.....	30
3.2.5 Úloha polohy mechanismu.....	33
3.2.6 Nejednoznačnost polohy mechanismu.....	34
3.2.7 Převodové funkce.....	35
3.3 Maticová metoda analýzy mechanismů.....	36
3.3.1 Poloha bodu.....	37
3.3.2 Sestavování transformační matice pomocí matic základních pohybů.....	38
3.3.3 Inverzní pohyb.....	40
3.3.4 Rychlost bodu a úhlové rychlosti tělesa.....	40
3.3.5 Zrychlení bodu a úhlové zrychlení tělesa.....	42
3.3.6 Matice zrychlení a rychlosti základních pohybů.....	43
3.3.7 Spojení více těles.....	44
3.3.8 Úloha polohy maticovou metodou.....	45
3.3.9 Rychlosti a zrychlení mechanismů.....	47
3.3.10 Numerické určování rychlostí a zrychlení mechanismu.....	49
4 Syntéza mechanismů.....	53
4.1 Formulace optimalizačního problému při syntéze mechanismů.....	54
4.2 Citlivostní analýza – montážní a výrobní tolerance.....	57
4.2.1 Výběr ladicích parametrů a jejich omezení.....	58
4.3 Optimalizace bez omezovacích podmínek.....	58
4.3.1 Metoda simplexů neboli metoda Nelder-Mead.....	59
4.3.2 Kvazínewtonovské metody.....	62
4.3.3 Srovnání metod.....	66
4.4 Optimalizace s omezovacími podmínkami.....	67
4.4.1 Metoda penalizačních funkcí.....	68
4.5 Lokální a globální minimum cílové funkce.....	72
4.6 Nesestavitelnost mechanismů během optimalizačního procesu.....	73

5 Metody řešení soustav nelineárních rovnic.....	77
5.1 Newtonova metoda.....	78
5.2 Broydenova dobrá metoda.....	81
5.3 Martinezova hybridní metoda.....	84
5.4 Trust-Region metody.....	86
5.5 Levenberg-Marquardt metoda.....	87
5.6 Genetické algoritmy.....	88
5.7 Podmíněnost řešení soustav nelineárních rovnic.....	89
5.8 Numerické testy metod pro řešení soustav nelineárních rovnic.....	90
5.8.1 Testovací soustava A - 18 rovnic.....	91
5.8.2 Testovací soustava B - 38 rovnic.....	92
5.8.3 Testovací soustava C - 2 rovnice.....	94
5.8.4 Testovací soustava D – nesestavitelný mechanismus.....	95
5.9 Srovnání metod pro řešení soustav nelineárních rovnic.....	98
6 Prohozní mechanismus tkacího stroje – dynamická analýza.....	101
6.1 Rozměry, zatížení a přijatá zjednodušení mechanismu.....	102
6.2 Úloha polohy.....	104
6.3 Rychlosti a zrychlení.....	106
6.4 Základní vztahy dynamiky mechanismů v maticovém vyjádření.....	107
6.5 Pohybové rovnice jednotlivých členů.....	109
6.6 Algoritmy výpočtu dynamické úlohy.....	112
6.6.1 Modifikovaná metoda ARV .....	112
6.6.2 Metoda Runge-Kutta 4. řádu.....	114
6.7 Srovnání výsledků dynamické úlohy.....	116
7 Levostranný mechanismus řazení.....	120
7.1 Rozdělení mechanismů řazení na levostranné a pravostranné.....	121
7.2 Souřadnice mechanismů řazení.....	121
7.3 Rozložení rychlostních stupňů převodové skříně ovládané levostranným mechanismem a hodnoty souřadnic mechanismu.....	122
7.4 Kinematický rozbor funkce levostranného mechanismu řazení.....	123
7.5 Stanovení počtu stupňů volnosti a počtu základních kinematických smyček.....	126
7.6 Sestavení maticových rovnic úlohy polohy levostranného mechanismu řazení.....	127
7.6.1 Maticová rovnice základní smyčky A.....	130
7.6.2 Maticová rovnice základní smyčky B.....	130
7.6.3 Maticová rovnice základní smyčky C.....	130
7.6.4 Shrnutí úlohy polohy.....	131
7.6.5 Postup výpočtu trajektorie bodu M.....	131
7.7 Tvar cílové funkce.....	132
7.7.1 Funkce kvantifikující vlastnosti jednotlivých drah řazení.....	134

7.7.2 Cílová funkce.....	136
7.8 Rozměry, dráhy řazení a citlivostní analýza levostranného mechanismu řazení.....	136
7.9 Optimalizace bez omezení levostranného mechanismu řazení.....	139
7.9.1 Dvourozměrná optimalizace parametrů L4s a L6s metodou simplexů.....	139
7.9.2 Dvourozměrná optimalizace parametrů L4s a L6s kombinovanou metodou BFGS.....	141
7.9.3 Trojrozměrná optimalizace parametrů L4s, L6s a $\psi$ metodou simplexů.....	143
7.9.4 Trojrozměrná optimalizace parametrů L4s, L6s a $\psi$ metodou BFGS.....	145
7.9.5 Trojrozměrná optimalizace bez omezení v multibody softwaru ANSYS Workbench 13.0.....	146
7.9.6 Trojrozměrná optimalizace bez omezení v multibody softwaru ADAMS 2011.....	150
7.9.7 Srovnání výsledků získaných autorem a multibody prostředí – optimalizace bez omezení.....	152
7.10 Optimalizace s omezením levostranného mechanismu řazení.....	153
7.10.1 Trojrozměrná optimalizace penalizační metodou.....	155
7.10.2 Trojrozměrná optimalizace provedená v multibody prostředích.....	156
7.10.3 Srovnání výsledků získaných autorem a multibody prostředí – optimalizace bez omezení.....	158
8 Pravostranný mechanismus řazení.....	159
8.1 Kinematický rozbor funkce pravostranného mechanismu řazení.....	160
8.2 Rozložení rychlostních stupňů převodové skříně ovládané pravostranným mechanismem a hodnoty souřadnic mechanismu.....	162
8.3 Stanovení počtu stupňů volnosti a počtu základních kinematických smyček.....	163
8.4 Maticové rovnice úlohy polohy pravostranného mechanismu řazení.....	164
8.4.1 Maticová rovnice základní smyčky I.....	164
8.4.2 Maticová rovnice základní smyčky II.....	166
8.4.3 Maticová rovnice základní smyčky III.....	166
8.4.4 Maticová rovnice základní smyčky IV.....	168
8.4.5 Maticová rovnice základní smyčky V.....	169
8.4.6 Maticová rovnice základní smyčky VI.....	170
8.4.7 Shrnutí úlohy polohy.....	171
8.4.8 Postup výpočtu trajektorie bodu M.....	171
8.5 Cílová funkce.....	172
8.5.1 Funkce jednotlivých drah řazení.....	173
8.6 Rozměry pravostranného mechanismu řazení.....	173
8.7 Dráhy řazení a srovnání matematického modelu mechanismu s hodnotami získanými z programu Pro/ENGINEER.....	174
8.8 Citlivostní analýza.....	176
8.8.1 Změna polohy kabiny ve směru $y_1$ .....	176
8.8.2 Naklopení osy válcové vazby.....	177
8.8.3 Shrnutí výsledků citlivostní analýzy.....	179
8.9 Alternativní provedení pravostranného mechanismu řazení.....	179
8.9.1 Variantní návrh A pravostranného mechanismu řazení.....	179
8.9.2 Variantní návrh B pravostranného mechanismu řazení.....	181
8.9.3 Alternativní provedení C pravostranného mechanismu řazení.....	182
9 Shrnutí dosažených výsledků a přínosů disertační práce.....	190

---

9.1 Metodika analýzy složitých prostorových mechanismů .....	190
9.2 Návrh matematické metody pro řešení úlohy polohy mechanismů.....	190
9.3 Implementace optimalizačních metod do úloh syntézy mechanismů.....	191
9.4 Problémy technické praxe.....	192
9.5 Algoritmy.....	194
9.6 Podněty k dalšímu výzkumu a práci.....	194
10 Conclusion.....	196
Seznam použitých pramenů.....	198
Seznam publikační činnosti doktoranda.....	202
Životopis doktoranda.....	203
Curriculum Vitae.....	204

## Obsah příloh

11 Přílohy.....	205
11.1 Procedury sestavující matice základních pohybů.....	205
11.2 Metoda BFGS, modifikovaná BFGS a DFP – numerické testy.....	206
11.2.1 Kombinovaná BFGS metoda.....	206
11.2.2 Numerické testy.....	206
11.3 Algoritmus metody simplexů neboli metody Nelder-Mead.....	209
11.4 Algoritmus metody BFGS.....	211
11.5 Metoda penalizace.....	212
11.6 Algoritmus Newtonovy metody řešení soustav nelineárních rovnic.....	213
11.7 Algoritmus Broydenovy dobré metody řešení soustav nelineárních rovnic.....	213
11.8 Algoritmus Martinezovy hybridní metody řešení soustav nelineárních rovnic.....	214
11.9 Prohozní mechanismus.....	216
11.10 Úloha polohy levostranného mechanismu řazení.....	216
11.10.1 Hlavní procedura.....	216
11.10.2 Procedura definující rozměry levostranného mechanismu.....	219
11.10.3 Procedura sestavující vektor hodnot reziduí algebraických rovnic získaných ze základních smyček.....	219
11.10.4 Procedura základní smyčky A.....	219
11.10.5 Procedura základní smyčky B.....	220
11.10.6 Procedura základní smyčky C.....	220
11.10.7 Optimalizace levostranného mechanismu.....	220
11.11 Úloha polohy pravostranného mechanismu řazení.....	222
11.11.1 Hlavní procedura.....	222
11.11.2 Procedura definující rozměry pravostranného mechanismu.....	225
11.11.3 Procedura sestavující vektor hodnot reziduí algebraických rovnic získaných ze základních smyček.....	226
11.11.4 Procedura základní smyčky I.....	227
11.11.5 Procedura základní smyčky II.....	227
11.11.6 Procedura základní smyčky III.....	227
11.11.7 Procedura základní smyčky IV.....	227
11.11.8 Procedura základní smyčky V.....	228
11.11.9 Procedura základní smyčky VI.....	228
11.12 Úloha polohy variantního návrhu A pravostranného mechanismu řazení.....	229
11.12.1 Hlavní procedura.....	229
11.12.2 Procedura definující rozměry pravostranného mechanismu variantního návrhu A.....	229
11.12.3 Procedura sestavující vektor hodnot reziduí algebraických rovnic získaných ze základních smyček.....	229
11.12.4 Procedura základní smyčky I.....	229
1.1.1 Procedura základní smyčky II.....	229
1.1.2 Procedura základní smyčky III.....	229
1.1.3 Procedura základní smyčky IV.....	230

1.1.4 Procedura základní smyčky V.....	230
1.1.5 Procedura základní smyčky VI.....	230
11.13 Úloha polohy variantního návrhu B pravostranného mechanismu řazení.....	231
11.13.1 Hlavní procedura variantního návrhu B.....	231
11.13.2 Procedura definující rozměry pravostranného mechanismu variantního návrhu B.....	231
11.13.3 Procedura sestavující vektor hodnot reziduí algebraických rovnic získaných ze základních smyček.....	231
11.13.4 Procedura základní smyčky I.....	231
11.13.5 Procedura základní smyčky II.....	231
11.13.6 Procedura základní smyčky III.....	232
11.13.7 Procedura základní smyčky IV.....	232
11.13.8 Procedura základní smyčky V.....	232
11.13.9 Procedura základní smyčky VI.....	232
11.14 Úloha polohy variantního návrhu C pravostranného mechanismu řazení.....	233
11.14.1 Hlavní procedura variantního návrhu C.....	233
11.14.2 Procedura definující rozměry pravostranného mechanismu variantního návrhu C.....	233
11.14.3 Procedura sestavující vektor hodnot reziduí algebraických rovnic získaných ze základních smyček.....	234
11.14.4 Procedura základní smyčky I.....	234
11.14.5 Procedura základní smyčky II.....	234
11.14.6 Procedura základní smyčky III.....	234
11.14.7 Procedura základní smyčky IV.....	235
11.14.8 Procedura základní smyčky V.....	235
11.15 Citlivostní analýzy pravostranného mechanismu řazení.....	235
11.15.1 Citlivost na posuv kabiny nákladního vozu.....	235
11.15.2 Změna polohy bodu S1.....	237
11.15.3 Změna polohy bodu V1.....	239
11.15.4 Změna polohy bodu RS1.....	239
11.16 Rešerše stavu poznání v problematice analýzy a syntézy mechanismů, matematické teorie optimalizace a řešení soustav nelineárních, algebraických rovnic.....	241
11.16.1 Analýza a syntéza mechanismů.....	241
11.16.2 Matematická teorie optimalizace a řešení soustav nelineárních, algebraických rovnic.....	246

## Seznam algoritmů

Algoritmus I – Metoda simplexů pro optimalizaci $n$ ladicích parametrů.....	56
Algoritmus II - Algoritmus kvazinevtonovských metod pro optimalizaci $n$ ladicích parametrů.....	59
Algoritmus III – Metoda penalizační funkce.....	67
Algoritmus IV – Algoritmus zobecňující metody řešení soustav nelineárních rovnic.....	73
Algoritmus V – Newtonova metoda pro řešení soustav nelineárních rovnic.....	76
Algoritmus VI – Broydenova dobrá metoda pro řešení soustav $n$ nelineárních rovnic.....	78
Algoritmus VII – Martinezova hybridní metoda.....	80
Algoritmus VIII - Modifikovaná metoda ARV pro dynamickou úlohu.....	109
Algoritmus IX - Subalgoritmus metody Runge-Kutta pro výpočet zrychlení.....	110
Algoritmus X - Algoritmus metody Runge-Kutta 4. řádu.....	111





## Seznam použitého značení

Při odkazování je použit následující systém - první číslo vždy označuje číslo kapitoly, ve které je zdroj publikovaný. Odkaz na číslovanou rovnici nebo vztah je uveden bez uvozovacího hesla, např. vypočteno dle 1.10. Při odkazech na obrázky číslo předchází zkratka obr., při odkazech na tabulky je použita zkratka tab. a pokud se jedná o odkaz na přílohu, tak použito slovo příloha.

Skalární veličiny jsou v práci značeny pomocí patkové kurzívy, například  $p$ . Vektory a matice jsou v práci značeny pomocí tučné, patkové kurzívy, například  $T$ . Vektory jsou značeny malými písmeny a matice velkými.

Pro tělesový souřadný systém je užívána zkratka TSS.

Velká písmena latinské abecedy		
Označení	Jednotka	Popis významu
$A$	-	Matice; matice zrychlení pohybu
$B$	-	Matice Hessián
	-	Vektor; bivektor
$C$	-	Označení „crawlingu“ neboli „plížení“
$D$	-	Matice; diferenciální operátor základního pohybu
$D_5$	$mm$	Vektor; souřadnice polohy bodu v souřadném systému 5
$DP$	-	Matice; derivace matice převodu transformační matice
$DV_4$	$mm$	Vektor; souřadnice polohy bodu v souřadném systému 4
$E$	-	Matice identity (jednotková)
$F$	-	Skalár; cílová funkce
$F$	$N$	Vektor; síla
$G$	-	Skalár; funkce, jenž kvantifikuje vlastnosti jednotlivých drah řazení
$H$	-	Inverzní matice k Hessiánu
	-	Inverzní matice k Jacobiho matici
$H_5$	$mm$	Vektor; souřadnice polohy bodu v souřadném systému 5
$HV_6$	$mm$	Vektor; souřadnice polohy bodu v souřadném systému 6
$I$	$kg\ m^2$	Skalár; moment setrvačnosti
$I$	$kg\ m^2$	Matice; tenzor setrvačnosti
$J$	-	Matice; Jacobiho matice
$L$	$mm$	Skalár; délka členu mechanismu
		Skalár; délka ojnice

**Velká písmena latinské abecedy**

Označení	Jednotka	Popis významu
$L_d$	$mm$	Skalár; délka drah řazení
$LS_{10}$	$mm$	Vektor; souřadnice polohy bodu v souřadném systému 10
$LV_1$	$mm$	Vektor; souřadnice polohy bodu v souřadném systému 1
$LV_6$	$mm$	Vektor; souřadnice polohy bodu v souřadném systému 6
$M$	-	Označení bodu rukojeti řadicí páky
$M_{12}$	$mm$	Vektor; souřadnice polohy bodu rukojeti řadicí páky v souřadném systému 12
$M$	$Nm$	Vektor; moment
$M_b$	$Nm$	Skalár; brzdny moment
$N$	-	Označení významných bodů volby
$O$	-	Vektor; souřadnice počátku souřadného systému
$O_1$	$mm$	Vektor; souřadnice polohy bodu v souřadném systému 1
$P$	-	Matice; matice převodu transformační matice
$PS_{10}$	$mm$	Vektor; souřadnice polohy bodu v souřadném systému 10
$PV_{11}$	$mm$	Vektor; souřadnice polohy bodu v souřadném systému 11
$Q$	-	Funkce; penalizovaná cílová funkce
$R$	- $mm$	Označení zpětného chodu Matice; průvodič působíště síly
$R_1$	$mm$	Vektor; souřadnice polohy bodu v souřadném systému 1
$R_{12}$	$mm$	Vektor; souřadnice polohy bodu v souřadném systému 12
$RK_6$	$mm$	Vektor; souřadnice polohy bodu v souřadném systému 6
$RS_1$	$mm$	Vektor; souřadnice polohy bodu v souřadném systému 1
$R_x$	-; $mm$	Transformační matice rotace kolem osy $x$
$R_y$	-; $mm$	Transformační matice rotace kolem osy $y$
$R_z$	-; $mm$	Transformační matice rotace kolem osy $z$
$S$	-	Matice; matice směrových kosinů
$S_1$	$mm$	Vektor; souřadnice polohy bodu v souřadném systému 1
$S_8$	$mm$	Vektor; souřadnice polohy bodu v souřadném systému 8
$SYM_1$	$mm$	Vektor; souřadnice polohy bodu v souřadném systému 1
$T$	-; $mm$	Transformační matice posuvu do bodu
$T_{ij}$	-; $mm$	Transformační matice
$T_x$	-; $mm$	Transformační matice posuvu ve směru osy $x$
$T_y$	-; $mm$	Transformační matice posuvu ve směru osy $y$
$T_z$	-; $mm$	Transformační matice posuvu ve směru osy $z$

**Velká písmena latinské abecedy**

Označení	Jednotka	Popis významu
$V$	-	Matice; matice rychlosti pohybu
$V_1$	$mm$	Vektor; souřadnice polohy bodu v souřadném systému 1

**Malá písmena latinské abecedy**

Označení	Jednotka	Popis významu
$a$	- -; $mm$	Skalár; konstanta rovnice přímky Skalár; prvek transformační matice.
$a$	$mm/s^2$	Vektor; rozšířený vektor zrychlení
$b$	- $mm$ $mm$ $Nms^2$	Skalár; konstanta rovnice přímky Skalár; vzdálenost kloubů Skalár; rozměr mechanismu Skalár; konstanta brzdného momentu
$c$	- -	Skalár; konstanta rovnice přímky Skalár; citlivost
$d$	$mm$	Skalár; křivost dráhy řazení Skalár; třída vazby
$div$	-	Skalár; hodnota sloužící k ukončení algoritmu.
$dp$	-	Skalár; derivace převodu mechanismu
$d$	$mm; ^\circ$	Vektor; krok kvazimewtonovských metod.
$d_{lim}$	$mm$	Skalár; rozměr zástavby vozidla
$e$	-	Vektor; bazový vektor
$f$	-	Funkce; funkce soustavy nelineárních rovnic
$g$	-	Funkce; nerovnostní omezující podmínka optimalizační úlohy
$g$	-	Vektor; gradient cílové funkce
$h$	$mm; ^\circ$ $mm$ -	Skalár; definuje okolí, na kterém je počítána numerická derivace Skalár; poloha pístu Funkce; nerovnostní omezující podmínka optimalizační úlohy
$h_{lim}$	$mm$	Skalár; rozměr zástavby vozidla
$i$	- - -	Skalár; pořadové číslo Skalár; počet stupňů volnosti mechanismu Symbolický index bodu dráhy řazení
$j$	- -	Skalár; pořadové číslo Symbolický index bodu dráhy řazení
$k$	-	Skalár; pořadové číslo iterace

**Malá písmena latinské abecedy**

Označení	Jednotka	Popis významu
	-	Skalár; pořadové číslo
	-	Skalár; počet vyčíslení
	-	Symbolický index dráhy řazení
<i>konv</i>	-	Skalár; výstupní proměnná algoritmu
<i>l</i>	<i>mm</i> ; °	Vektor; rozměry mechanismu
<i>m</i>	-	Fukce; kvadratický model cílové funkce
	-	Skalár; počet vazeb
	<i>kg</i>	Skalár; hmotnost
<i>maxit</i>	-	Skalár; maximální povolený počet iterací algoritmu
<i>n</i>	-	Skalár; počet dimenzí optimalizační úlohy
	-	Skalár; počet tělese mechanismu včetně rámu
<i>o</i>	-	Osa rotační vazby
<i>p</i>	-	Skalár; násobek jednotkové matice
	<i>mm</i>	Skalár; souřadnice mechanismu; posuv
	-	Skalár; převod mechanismu
	-	Skalár; vzdálenost
<i>p</i>	-	Vektor; Newtonův krok
<i>q</i>	<i>mm</i> ; °	Vektor; výstupní souřadnice mechanismu
	<i>mm</i> ; °	Vektor; neznámé souřadnice vazeb
<i>r</i>	<i>mm</i>	Rozšířený polohový vektor
	-	Vektor; rezidua v soustavě nelineárních rovnic
<i>r</i>	-	Skalár; počet základních řetězců
	-	Skalár; délka kliky
<i>s</i>	<i>mm</i>	Skalár; sklon dráhy řazení
<i>s</i>	-	Vektor; souřadnice mechanismu
<i>t</i>	-	Skalár; parametr BFGS metody
	<i>s</i>	Skalár; čas
<i>t</i>	<i>mm</i> ; °	Vektor; aritmetický průměr souřadnic vrcholů simplexu
<i>tol</i>	-	Skalár; poměrná tolerance vůči délce prvního kroku
<i>u</i>	-	Skalár; tolerance, se kterou počítá software
<i>v</i>	<i>mm</i> ; °	Vektor nových vrcholů simplexu
	<i>mm/s</i>	Vektor; rozšířený vektor rychlosti
<i>vule</i>	<i>mm</i>	Skalár; nezbytná vůle mezi členy mechanismu
<i>x</i>	<i>mm</i>	Skalár; souřadnice ve směru osy <i>x</i>
<i>x</i>	<i>mm</i> ; °	Vektor ladicích parametrů
	-	Vektor; neznámé v soustavě nelineárních rovnic

**Malá písmena latinské abecedy**

Označení	Jednotka	Popis významu
$\mathbf{x}_{min}$	$mm; ^\circ$	Vektor ladicích parametrů, který odpovídá minimu cílové funkce
$y$	$mm$	Skalár; souřadnice ve směru osy $y$
$\mathbf{y}$	- -	Vektor; rozdíl reziduálů Vektor; rozdíl gradientů
$z$	$mm$	Skalár; souřadnice ve směru osy $z$
$w_d$	-	Skalár; váhový faktor křivosti cílové funkce
$w_s$	-	Skalár; váhový faktor sklonu cílové funkce

**Velká písmena řecké abecedy**

Označení	Jednotka	Popis významu
$\Delta$	-	Skalár; poloměr oblasti důvěryhodnosti
$\Omega$	$s^{-1}$	Matice; matice úhlových rychlostí pohybu
$\dot{\Omega}$	$s^{-1}$	Matice; matice úhlových zrychlení pohybu

**Malá písmena řecké abecedy**

Označení	Jednotka	Popis významu
$\alpha$	$s^{-2}$	Vektor; úhlového zrychlení pohybu
$\alpha$	$^\circ$ -	Skalár; rozměr mechanismu Skalár; součinitel, který zpřesňuje délku kroku
$\beta$	$^\circ$	Skalár; rozměr mechanismu
$\gamma$	- $^\circ$	Skalár; parametr kontrakce metody simplexů Skalár; úhel sklonu osy rotační vazby spojující člen 5 a rám 1
$\delta$	$^\circ$	Skalár; úhel naklopení osy válcové vazby
$\epsilon_{BFGS}$	-	Skalár; hodnota sloužící k ukončení algoritmu metody BFGS
$\epsilon_{nes}$	-	Skalár; tolerance splnění úlohy polohy,
$\epsilon_{NS}$	-	Skalár; hodnota sloužící k ukončení výpočtu nelineárních soustav rovnic
$\epsilon_S$	-	Skalár; hodnota sloužící k ukončení algoritmu metody simplexů
$\kappa$	-	Skalár; číslo podmíněnosti
$\lambda$	-	Skalár; parametr BFGS metody
$\mu$	-	Skalár; penalizační parametr

## Malá písmena řecké abecedy

Označení	Jednotka	Popis významu
$\rho$	- -	Skalár; parametr zrcadlení metody simplexů Skalár; parametr BFGS metody
$\sigma$	-	Skalár; parametr zmenšení metody simplexů
$\varphi$	°	Skalár; souřadnice mechanismu; natočení
$\chi$	-	Skalár; parametr zvětšení metody simplexů
$\psi$	° °	Skalár; úhel sklonu osy rotační vazby Skalár; úhel sklonu křížového členu 3
$\omega$	$s^{-1}$	Vektor; úhlové rychlosti pohybu

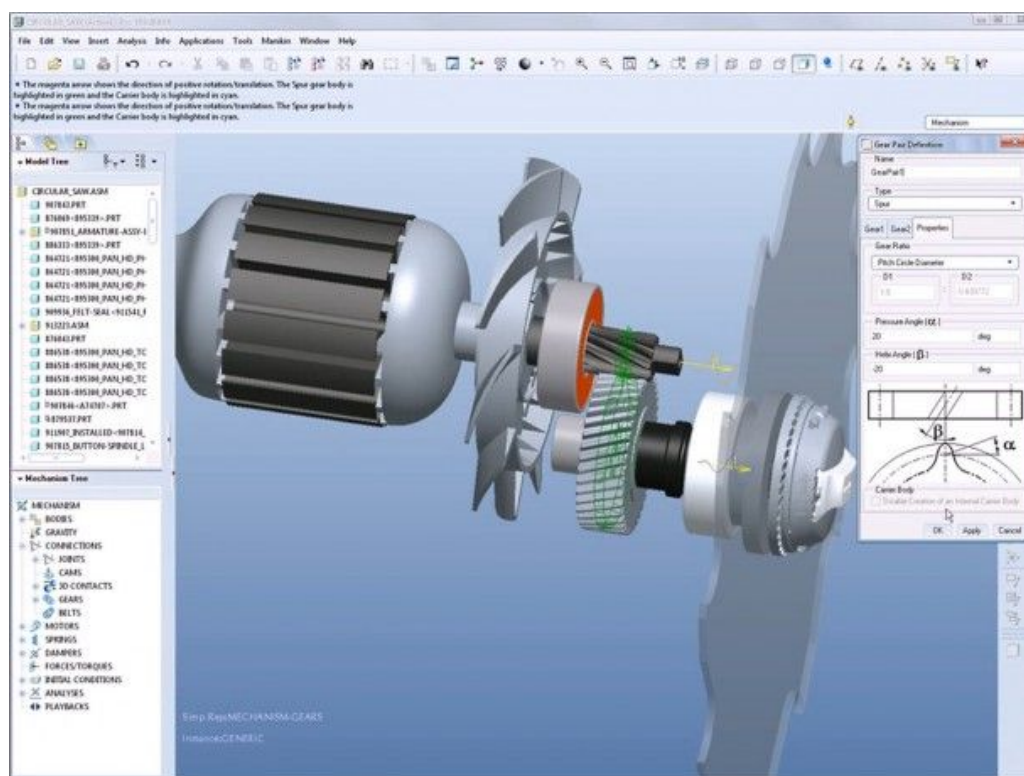
## Další značení

$\nabla$	-	Hamiltonův operátor
----------	---	---------------------

# 1 Úvod

Syntéza a analýza mechanismů se dnes provádí dvěma základními přístupy, a to buď použitím multibody systémů, nebo analyticky, kdy řešitel sám sestavuje matematický model reprezentující mechanismus.

Multibody systémem rozumíme softwarové prostředí, kde vytvoříme prostorový geometrický model mechanismu, viz obr. 1.1. Poté nadefinujeme vazby, kterými jsou tělesa spojena, a vstupní pohyb mechanismu. Na základě tohoto software automaticky sestaví matematický model a vyřeší jej. Výsledkem je video sekvence pohybu celého mechanismu. Taktéž lze sledovat vybrané veličiny - například posunutí vybraného bodu. Na trhu je dnes mnoho multibody systémů, z nichž lze uvést PTC Pro/ENGINEER, Adams, ANSYS, Simpack, DynaFlexPro, SAMCEF Mecano atd. Porovnání těchto softwarů lze najít například v [1] nebo [2].



Obrázek 1.1: Ukázka uživatelského rozhraní multibody systému

Druhou základní možností jak zkoumat vlastnosti mechanismů je analytický přístup, kdy řešitel sám sestavuje matematický model reprezentující mechanismus. Tento matematický model lze sestavit několika metodami, například vektorovou metodou, geometrickou metodou nebo metodou transformačních rovnic. Přehled metod lze najít ve [3], [4] nebo [5]. Pro komplikované, prostorové mechanismy je nejvhodnější využít metodu transformačních matic. Pokud je zapotřebí



se zabývat statikou nebo dynamikou prostorových mechanismů, je použití této metody téměř nezbytné.

Každý z těchto dvou přístupů má svoje klady i zápory, jejichž shrnutí je popsáno v tab. 1.1. Ze srovnání uvedeného v tabulce je zřejmé, že každý přístup je vhodný pro jiný typ práce s mechanismy. Multibody systémy je vhodné využít zvláště ve chvíli, kdy máme k dispozici prostorový model, máme již konkrétní představu o rozměrech mechanismu a potřebujeme rychle získat představu o vlastnostech mechanismu. Chceme-li podrobně zkoumat vlastnosti mechanismu je vhodné využít analytické metody. Nejvýznamnější výhodou analytických metod je přímý kontakt s matematickým modelem, takže lze získat jakoukoli veličinu. Naopak u multibody softwarů jsme omezeni tím, co nám programové prostředí dovoluje. Nezanedbatelné je také hledisko ceny softwaru. Ceny multibody softwarů se pohybují v řádech stovek tisíců korun. Při sestavení vlastního matematického modelu stačí mít k dispozici pouze základní výpočetní software, například tabulkový procesor, který lze pořídit i jako freeware.

*Tabulka 1.1: Srovnání multibody systémů a analytického přístupu*

	<i>Multibody systémy</i>	<i>Analytický přístup (konkrétně metoda transformačních matic)</i>
<i>Výhody</i>	Vytvoření simulace vyžaduje méně času (zvláště pokud je k dispozici prostorový model). Lepší vizualizace pohybu mechanismu. Zobrazení kolizí těles.	Model je efektivnější (výpočtové časy jsou nižší). Snadná parametrizace modelu. Jednoduchá syntéza.
<i>Nevýhody</i>	Vysoké pořizovací náklady na zakoupení licence softwaru. Často obtížná syntéza. Nepřímý kontakt s matematickým modelem. Pokud simulace pohybu havaruje, je často obtížné zjistit příčiny.	Více nároků na řešitele. Délka vytvoření simulace je silně závislá na schopnostech řešitele. Problematická vizualizace.

Tento text se dále soustředí na analytický přístup analýzy a syntézy mechanismů a jejich kinematických vlastností. Dynamikou mechanismů se zabývá práce pouze okrajově. Práce s multibody systémy je popsána pouze krátce a slouží hlavně pro srovnání výsledků.

## 2 Rešerše a cíle práce

Autor provedl rešerši podávající přehled o stavu poznání ve čtyřech oblastech, které jsou úzce spjatý s analýzou a syntézou mechanismů. Konkrétně jsou to tyto oblasti:

- analýza mechanismů
- syntéza mechanismů
- matematická teorie optimalizace
- řešení soustav nelineárních, algebraických rovnic

Průzkum v oblasti matematická teorie optimalizace byl proveden v návaznosti na syntézu mechanismů. A řešení soustav nelineárních, algebraických rovni je nezbytné při určování polohy mechanismů a tudíž při analýze mechanismů.

Tato podrobná rešerše je uvedena v příloze, viz 11.16.

### 2.1 Výsledky rešerše

Z provedené rešerše byly vyvozeny následující poznatky. Analýza kinematiky, statiky a dynamiky mechanismů je dnes obecně považována za dostatečně prozkoumanou oblast mechaniky tuhých těles. Autor však dospěl k názoru, že v oblasti analýzy prostorových mechanismů s vyšším počtem základních kinematických smyček jsou jisté nedostatky. Nejvhodnější metoda analýzy složitých mechanismů je metoda transformačních matic. Autor konkrétně postrádá metodiku, která by jasně a jednoduše popisovala postup analýzy těchto mechanismů pro širokou technickou veřejnost. Dostupná literatura popisující metodu transformačních rovnic je buď komplexní a zároveň pro běžnou technickou veřejnost nesrozumitelná, například [6], nebo příliš stručná, například [3] nebo [4]. V prozkoumaných zdrojích také zcela chybí nebo je zastaralý návod jak řešit úlohu polohy mechanismu, což je z pohledu matematiky problematika řešení soustav nelineárních rovnic. Z těchto závěrů plyne první cíl disertační práce, a to sestavit jasnou a srozumitelnou metodiku analýzy složitých prostorových mechanismů metodou transformačních matic přístupnou pro běžnou technickou veřejnost, a to včetně návodu a návrhu matematického řešení úlohy polohy. Tato metodika by měla být doplněna ukázkami na konkrétních příkladech.

Oblast syntézy mechanismů je detailně prozkoumána u analytických metod jako je například kolokační metoda. Souhrnným využitím optimalizačních algoritmů se zabývá Lederer, P. v [7] a [8], kniha je z roku 1989 a využívá často již zastaralé metody optimalizace. V [7] jsou

uvedeny příklady syntézy několika rovinných mechanismů. Jenže pouze u prvního příkladu jsou uvedeny numerické výsledky optimalizačního procesu. U ostatních příkladů je vždy uvedeno pouze kinematické schéma, kinematické popřípadě pohybové rovnice, formulace cílové funkce a vedlejších podmínek bez výsledku optimalizačního procesu. Zcela zde chybí aplikace nebo srovnání popsanych metod optimalizace na problematiku optimalizace mechanismů. Další zdroje jako [9], [10] nebo [11] se zabývají implementací optimalizačních algoritmů obecněji do problémů mechaniky, kde je hlavní pozornost většinou věnována hmotnostním optimalizačním poddajných těles při ohledu na dovolenou pevnost materiálu. Druhým cílem práce bude zkoumat implementaci optimalizačních algoritmů s omezením i bez omezení a samotnému nastavení úlohy syntézy mechanismů. Taktéž je nutné popsat možná úskalí průběhu optimalizačního procesu jako je například stav „Non-Assembly“ diskutovaný například v [12] nebo [13].

Z obou výše definovaných cílů vyplynula nutnost průzkumu stavu aktuálního vývoje matematické teorie optimalizace a řešení soustav nelineárních, algebraických rovnic. Dostupných materiálů a metod je mnoho a je zapotřebí tyto srovnat a vybrat metody vhodné pro užití v definovaných oblastech. Taktéž je nutné doporučit nastavení hodnot volitelných parametrů jednotlivých metod.

## **2.2 Cíle práce**

Navrhované téma disertační práce je analýza a syntéza mechanismů. Téma bylo navrženo na základě provedené rešerše stavu poznání ve čtyřech oblastech, a to analýza mechanismů, syntéza mechanismů, matematická teorie optimalizace a řešení soustav nelineárních, algebraických rovnic. Rešerše je uvedena v příloze 11.16. Z provedené rešerše bylo vyvozeno několik poznatků, které jsou detailně shrnuty v předchozí podkapitole. Tyto poznatky lze stručně shrnout následovně:

V prostudovaných zdrojích autor postrádá metodiku, která by srozumitelně popisovala postup analýzy prostorových mechanismů a byla tudíž přístupná široké technické veřejnosti, a to včetně návodu a návrhu matematického řešení úlohy polohy.

Oblast syntézy mechanismů je detailně prozkoumána u analytických metod, jako je například kolokační metoda. Naopak využití numerických optimalizačních metod pro syntézu mechanismů není v literatuře prezentováno dostatečně. Dostupné texty jsou buď zastaralé, nebo nepodávají komplexní popis využití optimalizačních metod.

Z obou výše uvedených závěrů rešerše vyplynula nutnost průzkumu stavu aktuálního

vývoje matematické teorie optimalizace a řešení soustav nelineárních algebraických rovnic. Dostupných materiálů a metod je mnoho a je zapotřebí tyto srovnat a vybrat metody vhodné pro užití v definovaných oblastech. Taktéž je nutné doporučit nastavení hodnot volitelných parametrů jednotlivých metod.

Z těchto poznatků odborné rešerše vyplynuly následující cíle disertační práce:

*1. Sestavit srozumitelnou metodiku analýzy složitých prostorových mechanismů metodou transformačních matic, která bude přístupná pro běžnou technickou veřejnost.*

*2. Navrhnout a popsat matematickou metodu pro řešení úlohy polohy.*

*3. Provést implementaci optimalizačních algoritmů s omezením i bez omezení do úloh syntézy mechanismů a tyto metody srovnat.*

*4. Aplikovat výše uvedené postupy a metody na konkrétních problémech technické praxe.*

### **3 Analýza kinematických vlastností prostorových mechanismů**

V této kapitole se budeme věnovat analýze kinematických vlastností mechanismů. Problematika je prezentována ve formě metodiky s důrazem na srozumitelnost a aplikovatelnost na problémy z technické praxe. Často je využito vysvětlení na jednoduchých příkladech z důvodů lepšího porozumění problematice.

#### **3.1 Základní pojmy teorie mechanismů**

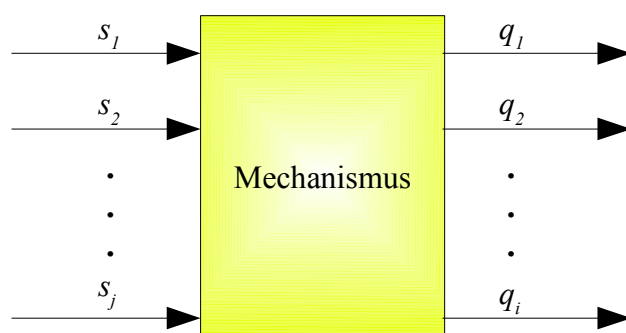
Abychom se mohli dále orientovat v textu, je nezbytné si osvojit tyto základní pojmy a přesně jim porozumět.

- Mechanismus - Pojmem mechanismus definujeme v technické mechanice soustavu těles, která je vzájemně spojena vazbami tak, aby přeměňovala jeden druh pohybu na jiný neboli vstupní pohyb na výstupní. Místo pojmu mechanismus je často používán také pojem soustava.
- Člen mechanismu - Jednotlivá tělesa, ze kterých je mechanismus složen, budou dále nazývána členy mechanismu. V textu bude věnována pozornost prostorovým mechanismům s tuhými členy. Pro člen mechanismu je taktéž využíván ekvivalent těleso mechanismu.
- Rám – Rám mechanismu je jedno nepohyblivé těleso, vůči němuž vyšetřujeme pohyby. Často se jedná o člen pevně spojený se zemí a je vždy označován indexem 1.
- Vstupní a výstupní člen – Vstupní člen je ten, jehož pohyb známe. Pohyb výstupního členu chceme zjistit.
- Vazba - Jednotlivé členy mechanismu jsou vzájemně spojeny vazbami, blíže kap. Chyba: zdroj odkazu nenalezen.
- Kinematická dvojice – Kinematickou dvojicí rozumíme dvě tělesa spojená vazbou.
- Zobecnělá souřadnice – Zobecnělou souřadnicí rozumíme délkovou nebo úhlovou souřadnici.
- Souřadnice vazby – Souřadnice vazby je hodnota (délka nebo úhel), která určuje vzájemnou polohu členů spojených vazbou. Každý mechanismus má několik vazeb, z nichž každá má několik souřadnic vazeb. Všechny souřadnice všech vazeb mechanismu budou řazeny ve

vektoru. Souřadnice vazby bude obecně značena písmenem  $\mathbf{q}$ .<sup>1</sup>

- Počet stupňů volnosti mechanismu - Počet stupňů volnosti mechanismu je počet vzájemně nezávislých pohybů, které může mechanismus vykonávat, viz kap. 3.2.3.
- Souřadnice mechanismu, též vstupní souřadnice – Jedna nebo více souřadnic, které jednoznačně určují polohu mechanismu. Jinými slovy lze říci, že souřadnice mechanismu určují polohu vstupního členu. Souřadnice mechanismu bude značena písmenem  $\mathbf{s}$ .<sup>1</sup>
- Výstupní souřadnice mechanismu – Výstupní souřadnicí mechanismu rozumíme polohu výstupního členu. Výstupní souřadnice mechanismu bude dále značena písmenem  $\mathbf{q}$ .<sup>1</sup> Všimněme si, že značení výstupní souřadnice a souřadnice vazby je shodné. Důvodem je, že výstupní souřadnicí je často poloha výstupního členu a poloha tohoto členu je právě dána souřadnicemi vazeb.
- Analýza kinematických vlastností mechanismu - Tímto pojmem rozumíme hledání závislosti poloh, rychlostí a zrychlení jednotlivých členů mechanismu na vstupním pohybu. Úlohou polohy rozumíme nalezení poloh všech členů mechanismu v závislosti na souřadnicích mechanismu. Vztah mezi souřadnicemi mechanismu (poloha vstupního členu) a výstupními souřadnicemi (poloha výstupního členu) nazýváme zdvihová závislost. Schematicky si lze představit mechanismus jako převodník, který transformuje vstupní souřadnice  $\mathbf{s}$  na výstupní souřadnice  $\mathbf{q}$ , viz obr. 3.1. Zdvihová závislost pro výstupní souřadnici  $q_i$  je funkcí všech souřadnic mechanismu  $\mathbf{s}$ , což lze matematicky zapsat jako

$$q_i = q_i(\mathbf{s}). \quad (3.1)$$

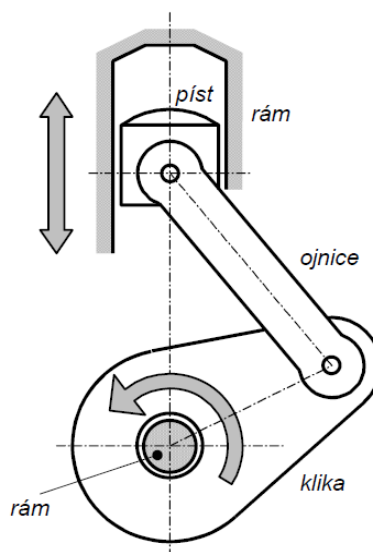


Obrázek 3.1: Schematické znázornění mechanismus jako převodníku

Výše uvedené pojmy lze vysvětlit na jednoduchém příkladě klikového mechanismu z obr. 3.2. Tento mechanismus je využíván například u spalovacích motorů nebo kompresorů. Funkce

<sup>1</sup> Tučné malé písmeno označuje, že veličina je vektor. Pro velikost vektoru bude užíván zápis  $a \in \mathbb{R}^m$ , což znamená, že vektor  $\mathbf{a}$  má  $m$  prvků, které jsou reálnými čísly.

tohoto mechanismu je následující: V prostoru válce se vznítí palivo, které vyvolá tlak na píst, který se začne pohybovat posuvným, přímočarým pohybem. Píst je kloubově spojen s ojnicí, která je dále kloubově spojena s klikou, která koná rotační pohyb. Dle definice mechanismů uvedené v prvním odstavci klikový mechanismus převádí přímočarý pohyb na pohyb rotační. Členy mechanismu jsou rám, píst, ojnice a kliky. Funkci rámu má válec a ukotvení kliky. Vstupním členem je píst a výstupním je kliky. Počet vazeb mechanismu je 4, například kliky je spojena s ojnicí rotační vazbou jejíž souřadnice je úhel natočení. Počet stupňů volnosti mechanismu je roven jedné, protože mechanismus vykonává pouze jeden nezávislý pohyb. Souřadnice mechanismu neboli vstupní souřadnice je jedna a je to poloha pístu. Výstupní souřadnicí je natočení kliky. Souřadnice vazeb jsou čtyři, a to posunutí pístu a natočení třech rotačních vazeb, z nichž posunutí pístu odpovídá vstupní souřadnici zatímco natočení rotační vazby mezi rámem a klikou odpovídá výstupní souřadnici. Tímto se potvrzuje, že výstupní souřadnice je často shodná s některou souřadnicí vazby. Analýzou kinematických vlastností klikového mechanismu bychom získali vztahy pro polohu, rychlost a zrychlení ojnice a kliky v závislosti na poloze, rychlosti a zrychlení pístu. Zdvihová závislost by definovala vztah mezi posunutím pístu a natočením kliky.



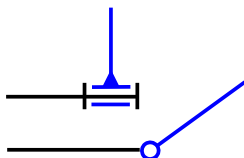
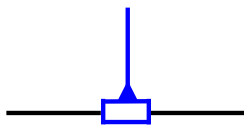
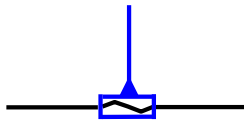
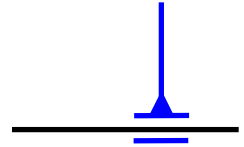
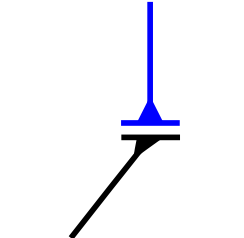
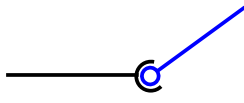
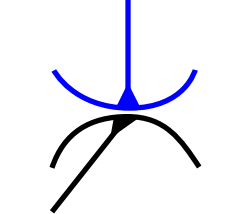
*Obrázek 3.2: Klikový mechanismus; převzato z [55]*

## 3.2 Vlastnosti mechanismů

V této kapitole shrneme a vysvětlíme vlastnosti mechanismů, které je nutno znát pro jejich následnou analýzu.

### 3.2.1 Prostorové vazby a jejich schematické značky

Tabulka 3.1: Rozdělení vazeb dle třídy a jejich schematické značky

Typ vazby	Počet a druh stupňů volnosti vazby	Třída vazby $d$ (počet odebraných stupňů volnosti vazbou)	Schematická značka	Označení vazby písmenem	
Rotační	1 rotace	5		R	Nižší vazby
Posuvná	1 posuv	5		P	
Šroubová	1 posuv svázan s rotací	5		Š	
Válcová	2 1 posuv, 1 rotace	4		V	
Rovinná	3 2 posuvy, 1 rotace	3		F	
Sférická (kloubová)	3 3 rotace	3		S	
Obecná (vazba s pěti stupni volnosti)	5 3 rotace, 2 posuvy	1		G	Vyšší vazby



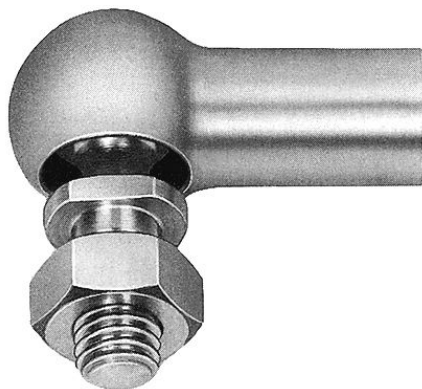
Vazba je prvek, který spojuje dva členy mechanismu neboli vazba určuje, jakým vzájemným pohybem se tyto dva členy mohou pohybovat. Dva členy spojené vazbou nazýváme kinematickou dvojicí. Často rozlišujeme vazby ideální, kde jsou zanedbány pasivní odpory (tření), a reálné, kde jsou pasivní odpory zahrnuty.

Dále dělíme prostorové vazby na:

- Nižší vazby - spojené členy se dotýkají v ploše; např. rotační nebo posuvná vazba.
- Vyšší vazby – spojené členy se dotýkají v křivce nebo v bodě; například styk dvou tyčí (dotyk v bodu).

Nutno poznamenat, že vyšetřování mechanismů, které obsahují vyšší kinematické dvojice s sebou nese jisté komplikace v porovnání s mechanismy obsahujícími nižší kinematické dvojice.

Dalším podstatným dělením vazeb je dle počtu stupňů volnosti, které vazba odebírá; tento počet odebraných stupňů volnosti se nazývá třída vazby a je značen  $d$ . Rozdělení vazeb dle třídy je uvedeno v tab. 3.1, dále je zde rozpracováno schématické značení vazeb dle ČSN EN ISO 3952-1-2-3-4. Na obr. 3.3 lze vidět reálný vzhled sférické neboli kloubové vazby, která je v kinematických schématech značena dle tab. 3.1.

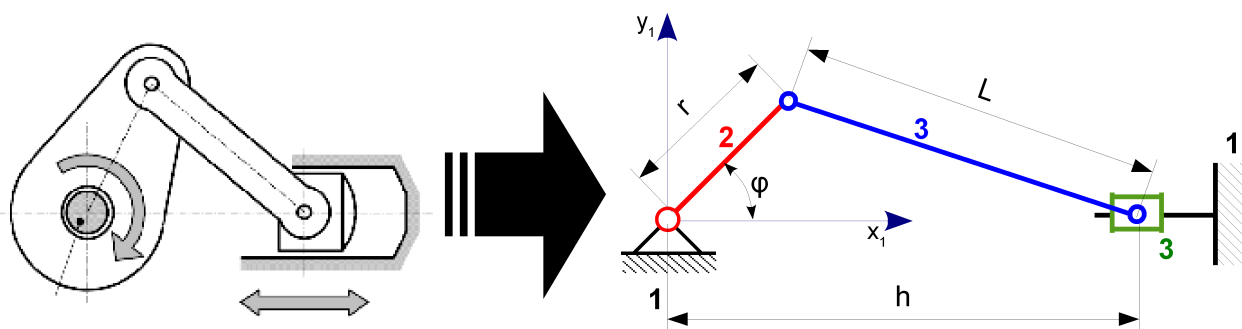


*Obrázek 3.3: Reálný vzhled  
sférické neboli kloubové vazby*

### **3.2.2 Kinematické schéma mechanismu**

Prvním krokem při vyšetřování jakýchkoliv vlastností mechanismů je sestavení přehledného kinematického schématu. Kinematické schéma je co nejvíce zjednodušené schéma mechanismu, které zachovává pouze pro kinematiku popřípadě dynamiku podstatné rozměry (většinou polohy vazeb nebo zatížení), zavádí schématické značení vazeb dle tab. 3.1 a v případě

dynamiky silové účinky apod. Současně je třeba zavést číslování těles. Číslem 1 se vždy značí rám. Tělesa je vhodné od sebe odlišovat barevně, což značně ulehčí orientaci zvláště u složitějších mechanismů. Při popisování mechanismu v textu je vhodné vždy při uvedení tělesa přidat i jeho pořadové číslo, např. tyč 6. Dále je vhodné přiřadit základní (někdy též nazýván globální) souřadný systém příslušející pevnému prostoru neboli rámu, který je označen pořadovým číslem 1. Příklad jednoduchého kinematického schématu je ukázán na klikovém mechanismu uvedeném v obr. 3.4. Další kinematická schémata komplikovanějších mechanismů lze nalézt na obr. 7.6 a 8.4.



Obrázek 3.4: Kinematické schéma klikového mechanismu

### 3.2.3 Počet stupňů volnosti mechanismu

Základní vlastností mechanismu je počet stupňů volnosti, což je počet vzájemně nezávislých pohybů, které může mechanismus vykonávat, nebo jinými slovy je to nejmenší počet nezávislých souřadnic mechanismu, které jednoznačně určují jeho polohu.

U klikového mechanismu z obr. 3.4 je jeho počet stupňů volnosti roven jedné, což znamená, že existuje jedna souřadnice, která jednoznačně určuje jeho polohu a touto souřadnicí je úhel natočení kliky  $\varphi$ . Zablokujeme-li myšleně tuto souřadnici, nemůže již mechanismus vykonávat žádný pohyb. Pokud by mechanismu mohl dále vykonávat pohyb, znamenalo by to, že soustava má více stupňů volnosti.

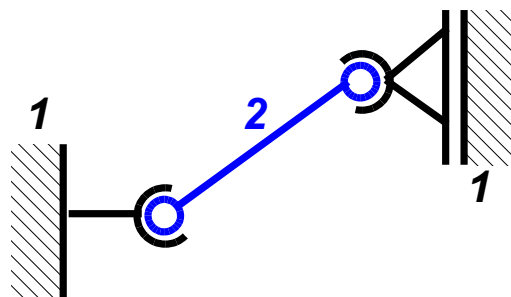
Počet stupňů volnosti mechanismu  $i$  stanovujeme v technické praxi dle vztahu

$$i = 6 \cdot (n - 1) - \sum_{j=1}^m d_j, \quad (3.2)$$

kde  $n$  je počet členů mechanismu včetně rámu,  $m$  počet vazeb a  $d_j$  je třída vazby viz kap. 3.2.1.

Počet stupňů volnosti  $i$  je většinou roven počtu vstupních souřadnic, které jsou řazeny ve vektoru  $s$ . Výjimkou jsou mechanismy, kde existuje člen s tzv. volným stupněm volnosti. Například tyč, která je na obou koncích uchycena sférickým kloubem, může neomezeně rotovat kolem své osy,

viz obr. 3.5. Pokud by se v mechanismu taková tyč nacházela, tak by počet stupňů volnosti  $i$  byl roven počtu vstupních souřadnic plus jeden volný stupeň volnosti (rotace) tyče. Toto se ošetří při řešení úlohy polohy předepsáním nulového úhel, který definuje natočení tyče, viz kap. 7.6.2.



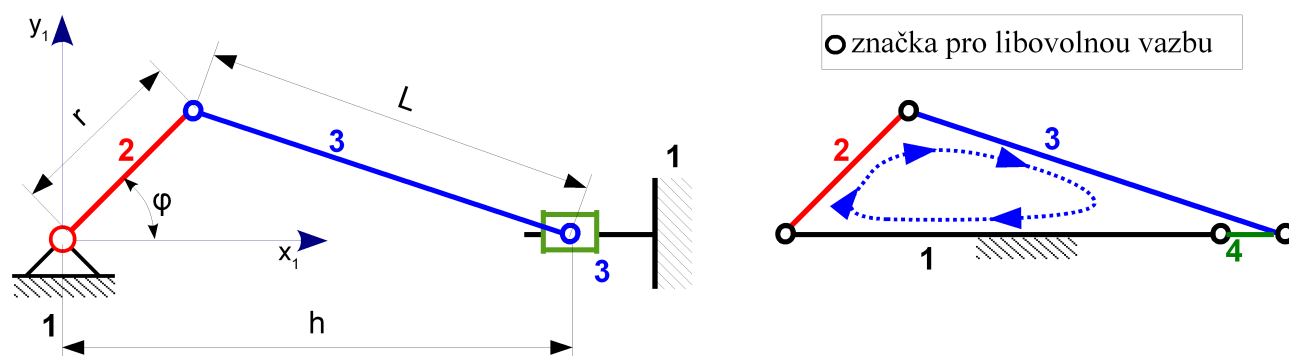
Obrázek 3.5: Ukázka volného stupně volnosti - tyč může rotovat kolem své osy

Vztah 3.2 neplatí pro tzv. výjimečné případy mechanismů, což jsou soustavy, které jsou z důvodu specifického vzájemného uspořádání členů nebo jejich rozměrů nepohyblivé, například obr. 4.8 a 4.9. Existuje teorie, která o tomto pojednává a ošetřuje. Tyto příklady se vyskytují velice zřídka a proto zde není tato problematika dále rozvedena, pro bližší seznámení viz [5] str. 355.

### 3.2.4 Kinematické řetězce a smyčky

Další významnou charakteristikou mechanismů je jeho topologie (jinými slovy rozvětvenost). Abychom mohli kvantifikovat jak dalece je mechanismus rozvětvený, zavedeme pojem kinematický řetězec. Kinematický řetězec získáme, spojíme-li několik těles vazbami.

Pro účely zkoumání kinematických řetězců se osvědčilo zavedení zjednodušeného kinematického schématu, a to tak, že jsou odstraněny rozměry a místo vazeb jsou zakresleny neurčité kroužky dle obr. 3.6, kde je toto provedeno pro klikový mechanismus. Je vhodné označit místa, kde je soustava vázaná na rám 1. Ze schématu vyplývá, že klikový mechanismus je tvořen jedním kinematickým řetězcem. Dále si lze všimnout, že tento řetězec je uzavřený do tzv. kinematické smyčky, což graficky naznačuje uzavřená křivka s šipkami.



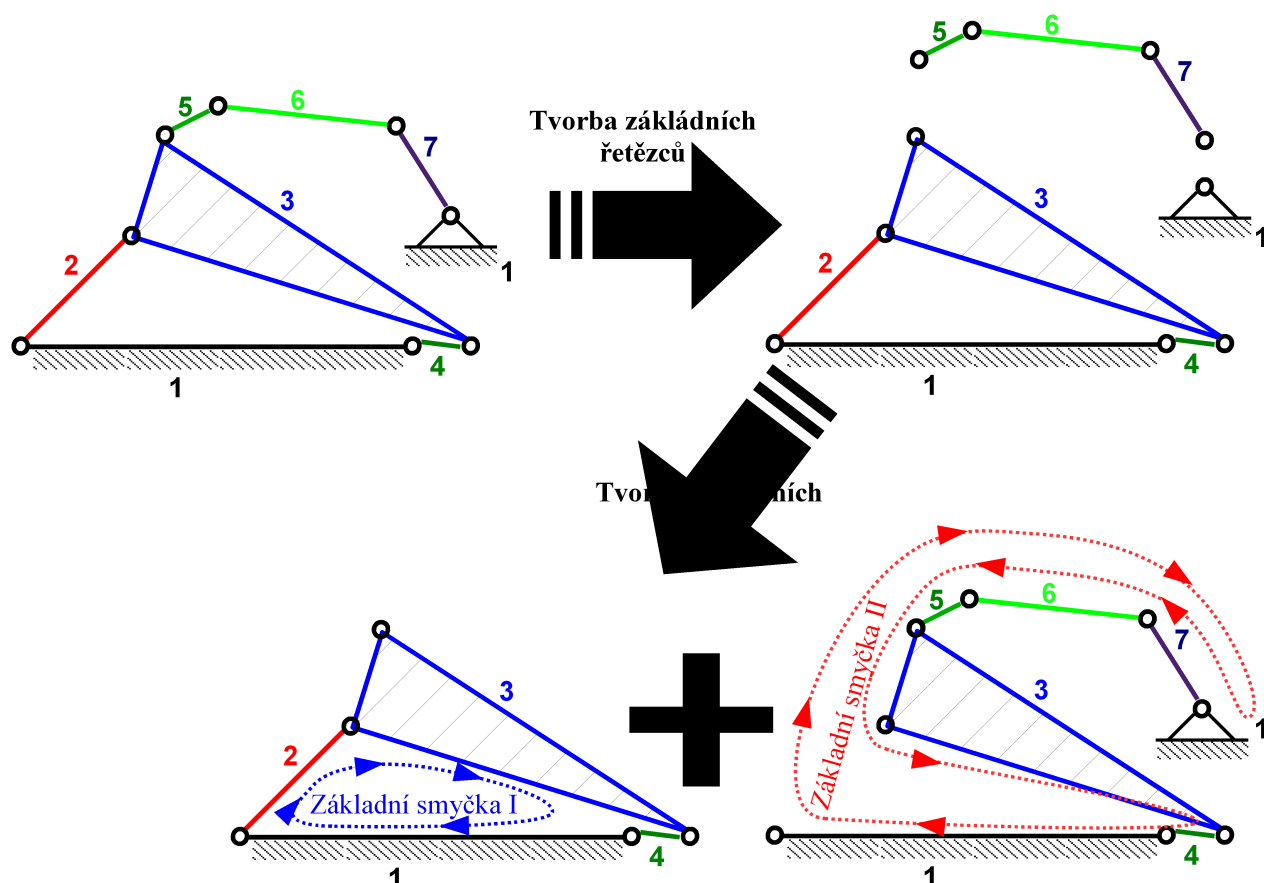
Obrázek 3.6: Zavedení zjednodušeného kinematického schématu

Pro další výklad je nutné kinematické řetězce rozdělit podle dvou hledisek dle následující tab. 3.2.

Tabulka 3.2: Rozdělení kinematických řetězců

<i>Rozdělení řetězců na uzavřené, otevřené a smíšené.</i>		
	<i>Popis</i>	<i>Příklad schématu</i>
Uzavřený kinematický řetězec (kinematická smyčka)	Členy mechanismu tvoří smyčku; např. klikový mechanismus.	
Otevřený kinematický řetězec	Např. robotické rameno.	
Smíšený kinematický řetězec		
<i>Rozdělení řetězců na jednoduché a složené</i>		
	<i>Popis</i>	<i>Příklad schématu</i>
Jednoduchý kinematický řetězec	Každý člen mechanismu je napojen nejvíce na dvě tělesa	
Složený kinematický řetězec	Některý ze členů mechanismu je napojen na více než dvě tělesa	

Otevřené řetězce, kterým zde je věnována pozornost pouze okrajově, se vyskytují zvláště při konstrukci robotických ramen a podobných zařízení.



Obrázek 3.7: Postup tvorby základních smyček

Nejčastěji zkoumáme mechanismy, které jsou tvořeny smíšeným kinematickým řetězcem, kde jednotlivé řetězce jsou uzavřené. Tyto soustavy se vyšetřují pomocí teorie tzv. základních smyček. Jako příklad uveďme mechanismus, jehož zjednodušené schéma je uvedeno na obr. 3.7 vlevo nahoře. Dle tab. 3.2 lze říci, že soustava se skládá ze smíšeného kinematického řetězce a dále lze konstatovat, že zde není žádný otevřený řetězec. Nyní tento mechanismus rozpojíme v jeho vazbách tak aby vznikl jeden jednoduchý uzavřený řetězec a dále pouze jednoduché otevřené řetězce. Toto je provedeno na obr. 3.7 vpravo nahoře, kde uzavřený jednoduchý řetězec se skládá z členů 1, 2, 3, 4 a otevřený jednoduchý řetězec je jeden, který se skládá z členů 5, 6, 7. Takto vzniklé kinematické řetězce nazýváme základní kinematické řetězce. Jejich počet je pro jeden určitý mechanismus neměnný, avšak jejich uspořádání je volbou řešitele. Počet základních smyček bude označován písmenem  $r$ . Například pro tento konkrétní příklad by mohlo být alternativní uspořádání následující: členy 1, 2, 3, 5, 6, 7 tvoří jednoduchý uzavřený řetězec a člen 4 jednoduchý otevřený řetězec. Uspořádání řetězců je jiné, ale jejich počet dvě je stále stejný. Nyní vzniklé jednoduché otevřené řetězce doplníme ostatními členy tak, aby tvořily smyčku. Vzniknou nám tzv. základní smyčky jejichž počet je roven počtu základních řetězců  $r$ , viz obr. 3.7 dole. Základní smyčka I se skládá z členů 1, 2, 3, 4 a základní smyčka II se skládá z členů 1, 7, 6, 5, 3, 4. Pokud bychom vyšetřovali pohyb mechanismu jako na obr. 3.7, tak

bychom tvořili úlohu polohy právě pro každou základní smyčku zvlášť a tolikrát, kolik je základních smyček; proto je nutné znát celý proces, který vede až k základním smyčkám. Tvorbu základních smyček pro složitější mechanismy lze najít v kap. 7.5 a 8.3.

Pro přehlednost definujeme postup, který shrnuje předcházející odstavec. Postup platí pro zkoumání soustav tvořených smíšeným kinematickým řetězcem bez otevřených řetězců.

Postup tvorby základních smyček: Schéma mechanismu rozdělíme na jeden uzavřený jednoduchý řetězec a další otevřené jednoduché řetězce; všechny tyto řetězce (uzavřené i otevřené) označujeme jako základní řetězce a jejich počet značíme  $r$ . Vzniklé otevřené základní řetězce doplníme ostatními členy tak, aby tvořily smyčku. Tyto vzniklé smyčky a uzavřený základní řetězec nazýváme základní smyčky a jejich počet je stále  $r$ .

Další příklady tvorby základních smyček lze nalézt v [3] na str. 32 nebo [6] na str. 38.

### 3.2.5 Úloha polohy mechanismu

Úloha polohy slouží ke stanovení poloh jednotlivých členů mechanismu a k určení výstupních souřadnic v závislosti na vstupních souřadnicích. Primárně z úlohy polohy získáme všechny souřadnice vazeb, ze kterých lze určit polohy těles a výstupní souřadnice mechanismu.

Například u klikového mechanismu z obr. 3.6, kde vstupní souřadnicí je poloha pístu  $h$ , úloha polohy stanovuje polohy jednotlivých členů pomocí hodnot natočení jednotlivých rotačních vazeb a dále stanovuje hodnotu výstupní souřadnice  $\varphi$ , což je natočení kliky.

Uvážíme-li obecný případ prostorového mechanismu tvořeného  $r$  základními řetězci, má úloha polohy z matematického pohledu formu soustavy nelineárních, transcendentních a skalárních rovnic. Počet rovnic této soustavy vyplývá z počtu základních smyček  $r$  a je roven

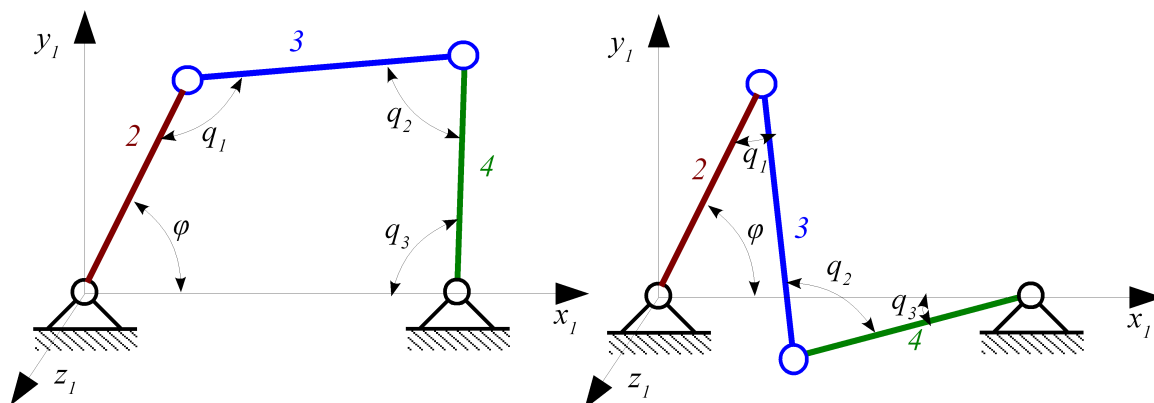
$$r \cdot 6. \quad (3.3)$$

Tuto soustavu rovnic vyjadřující úlohu polohy lze zapsat následovně

$$\begin{aligned} f_1(\mathbf{s}, \mathbf{l}, \mathbf{q}) &= 0, \\ f_2(\mathbf{s}, \mathbf{l}, \mathbf{q}) &= 0, \\ &\vdots \\ f_{r \cdot 6}(\mathbf{s}, \mathbf{l}, \mathbf{q}) &= 0, \end{aligned} \quad (3.4)$$

kde  $\mathbf{s}$  označuje vektor vstupních souřadnic,  $\mathbf{l}$  vektor rozměrů mechanismu a  $\mathbf{q}$  vektor neznámých souřadnic vazeb (úhly natočení nebo posunutí). U jistých mechanismů bývá počet neznámých souřadnic vazeb vyšší než počet rovnic soustavy. Jedná se o případ, který je zobrazen na obr. 3.5, kde je také vysvětlující text. Postup získání soustavy 3.4 je uveden v kapitole 3.3.8.

### 3.2.6 Nejednoznačnost polohy mechanismu



Obrázek 3.8: Nejednoznačnost polohy čtyřkloubového mechanismu

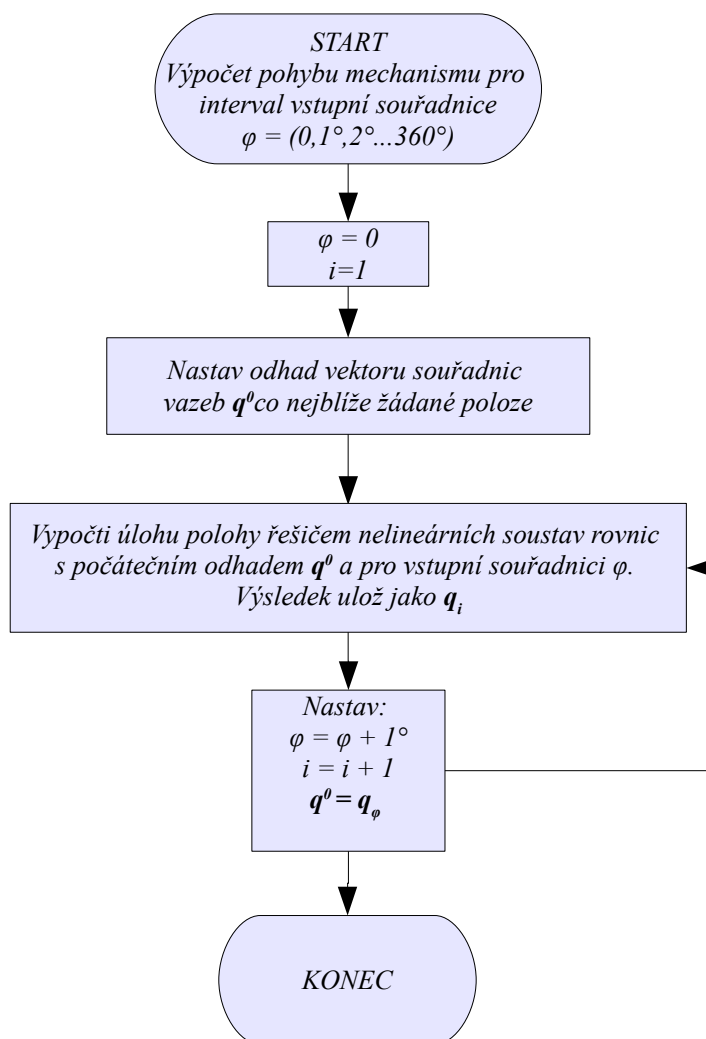
Nejednoznačností polohy mechanismu rozumíme, že pro jeden vektor vstupních souřadnic existuje několik variant polohy mechanismu neboli několik vektorů souřadnic vazeb  $\mathbf{q}$ . Toto je názorně ukázáno na obr. 3.8, kde pro jednu hodnotu vstupní souřadnice  $\varphi$  čtyřkloubového mechanismu existují dvě varianty polohového uspořádání. Jinými slovy, vstupní souřadnici  $\varphi$  odpovídají dva vektory souřadnic vazeb  $\mathbf{q}$ , které definují polohu mechanismu.

Tato nejednoznačnost polohy může způsobovat potíže při řešení úlohy polohy, která je řešena některou z numerických metod viz kap. 5. Pro nejednoznačné mechanismy má tato soustava více řešení. Této komplikaci se lze v technické praxi jednoduše vyhnout tím, že při vyhodnocování soustavy 3.4 zadáme počáteční odhad vektoru  $\mathbf{q}^0$  tak, aby odpovídal co nejbližší námi žádané poloze (toto lze provést odhadem nebo měřením z kreslicího softwaru).

Při studiu pohybu mechanismu neboli pokud chceme znát pohyb mechanismu v rámci určitého intervalu vstupní souřadnice, postupujeme následujícím způsobem, který popíšeme na jednoduchém případu z obr. 3.8. Analyzujeme čtyřkloubový mechanismus pro tento interval vstupní souřadnice

$$\varphi \in \langle 0^\circ, 1^\circ, 2^\circ, \dots, 360^\circ \rangle. \quad (3.5)$$

Během analýzy úlohy polohy by se mohlo stát, že pro jednu hodnotu vstupní souřadnice bude soustava uspořádána jako na obr. 3.8 vlevo a hned pro další hodnotu vstupní souřadnice z intervalu 3.5 by mechanismus „přeskočil“ do polohy jako na obr. 3.8 vpravo, což zřejmě neodpovídá realitě. Tomuto přeskokování zabráníme opět pomocí správného počátečního odhadu souřadnic vazeb  $\mathbf{q}^0$ . První odhad provedeme měřením, jak již bylo popsáno, a další odhady nastavujeme jako výsledky minulých úloh poloh, viz vývojový diagram na obr. 3.9.



Obrázek 3.9: Vývojový diagram analýzy pohybu mechanismu zabraňující „přeskakování“

### 3.2.7 Převodové funkce

Uvažujme zdvihovou závislost popsanou v kap. 3.1 pro zobecnělou výstupní souřadnici  $q_1$ , která je funkcí všech souřadnic mechanismu seřazených ve vektoru  $s$ , což lze matematicky zapsat jako

$$q_1 = q_1(s). \quad (3.6)$$

Potom zobecnělou výstupní rychlost  $\dot{q}_1$  lze vypočíst derivací podle času a s využitím totálního diferenciálu získáme vztah

$$\dot{q}_1 = \frac{d q_1(s)}{dt} = \frac{\partial q_1}{\partial s_1} \cdot \dot{s}_1 + \frac{\partial q_1}{\partial s_2} \cdot \dot{s}_2 + \dots + \frac{\partial q_1}{\partial s_i} \cdot \dot{s}_i, \quad (3.7)$$

kde  $\dot{s}_i$  je zobecnělá vstupní rychlost. Vztah 3.7 lze zapsat jako

$$\dot{q}_1 = p_{1,1} \cdot \dot{s}_1 + p_{1,2} \cdot \dot{s}_2 + \dots + p_{1,i} \cdot \dot{s}_i, \quad (3.8)$$



kde  $p$  se nazývá převod, jehož dolní index před čárkou značí výstupní souřadnici, kterou derivujeme, a index za čárkou značí podle které vstupní souřadnice převod derivujeme.

Podobným způsobem lze získat vztah pro zrychlení první zobecnělé výstupní souřadnice  $\ddot{q}_1$

$$\ddot{q}_1 = p_{1,1}^2 \cdot \ddot{s}_1 + p_{1,2}^2 \cdot \ddot{s}_2 + \frac{\partial p_{1,1}}{\partial s_1} \cdot \dot{s}_1^2 + \frac{\partial p_{1,2}}{\partial s_2} \cdot \dot{s}_2^2 + 2 \cdot \dot{s}_1 \cdot \dot{s}_2 \cdot \frac{\partial p_{1,1}}{\partial s_2}, \quad (3.9)$$

kde přibýly oproti vztahu pro rychlosti tzv. derivace převodu, které budou v textu často značeny jako  $dp$  s tím, že index před čárkou značí derivovaný převod a indexy za čárkou značí podle kterých vstupních souřadnic převod derivujeme. Pokud bude mít mechanismus pouze jednu vstupní souřadnici, tak u  $dp$  budou indexy za čárkou vynechány.

Vztah 3.9 lze přepsat

$$\ddot{q}_1 = p_{1,1}^2 \cdot \ddot{s}_1 + p_{1,2}^2 \cdot \ddot{s}_2 + dp_{1,11} \cdot \dot{s}_1^2 + dp_{1,22} \cdot \dot{s}_2^2 + 2 \cdot \dot{s}_1 \cdot \dot{s}_2 \cdot dp_{1,12}. \quad (3.10)$$

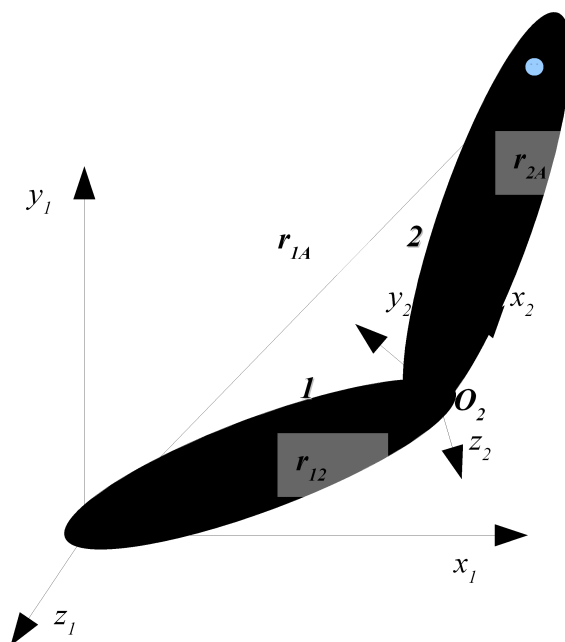
Detailnější popis převodových funkcí lze nalézt v [55]. Vztahy 3.8 a 3.9 nám definují postup jak získat rychlost a zrychlení kterékoli zobecnělé souřadnice nebo souřadnice vazby pomocí vstupní souřadnice a její rychlosti a zrychlení a použitím převodových funkcí. tento postup bude dále využit v kap. 3.3.10.

### 3.3 Maticová metoda analýzy mechanismů

Pro analýzu úlohy polohy rychlostí a zrychlení mechanismů byla vyvinuta celá řada metod. Mezi nejznámější metody patří trigonometrická, vektorová nebo maticová metoda apod. viz [14]. Dále lze nalézt i méně známé metody jako metoda generovaných povrchů, metoda rovnic konstantních vzdáleností atd, viz [3] od strany 103.

Pro mechanismy s více základními smyčkami a prostorové mechanismy je jednoznačně nejvhodnější maticová metoda, kterou v této kapitole popíšeme a která stanoví postup získání úlohy polohy a vyšetření rychlostí a zrychlení mechanismu.

### 3.3.1 Poloha bodu



Obrázek 3.10: Tělesa spojená libovolnou s tělesovými souřadnými systémy

Základním stavebním kamenem maticové analýzy mechanismů je tzv. maticová transformační rovnice, kterou následně popíšeme a vyjdeme přitom z obr. 3.10. Představme si dvě tělesa, která jsou spojena libovolnou vazbou dle kap. Chyba: zdroj odkazu nenalezen a tato tělesa označme pořadovými čísly 1 (modře) a 2 (červeně). Těmto tělesům přiřadíme tzv. tělesové souřadné systémy<sup>2</sup>  $x_1, y_1, z_1$  a  $x_2, y_2, z_2$ , které jsou pevně spojeny s těmito tělesy. Jinými slovy, pokud se pohybuje těleso 1, pohybuje se s ním současně i TSS 1. Tyto systémy musí být nezbytně pravotočivé.

Nyní se zaměříme na bod  $A$ , který je součástí tělesa 2, a cílem je popsat polohu tohoto bodu v TSS 1. Toto lze provést maticovou transformační rovnicí, která má následující tvar

$$\mathbf{r}_{1A} = \mathbf{T}_{12} \cdot \mathbf{r}_{2A}, \quad (3.11)$$

kde jednotlivé členy maticové transformační rovnice mají následující strukturu

$$\mathbf{r}_{1A} = \begin{bmatrix} x_{1A} \\ y_{1A} \\ z_{1A} \\ 1 \end{bmatrix}, \quad \mathbf{r}_{2A} = \begin{bmatrix} x_{2A} \\ y_{2A} \\ z_{2A} \\ 1 \end{bmatrix} \quad \text{a} \quad \mathbf{T}_{12} = \begin{bmatrix} \cos \langle x_1, x_2 \rangle & \cos \langle x_1, y_2 \rangle & \cos \langle x_1, z_2 \rangle & x_{O2} \\ \cos \langle y_1, x_2 \rangle & \cos \langle y_1, y_2 \rangle & \cos \langle y_1, z_2 \rangle & y_{O2} \\ \cos \langle z_1, x_2 \rangle & \cos \langle z_1, y_2 \rangle & \cos \langle z_1, z_2 \rangle & z_{O2} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.12)$$

Vektor  $\mathbf{r}_{2A}$  se nazývá rozšířeným polohovým vektorem a definuje polohu bodu  $A$  v TSS

<sup>2</sup> Pro tělesový souřadný systém bude používána zkratka TSS.

<sup>3</sup> Matice budou značeny velkým tučným písmenem.

2. Prvky tohoto vektoru jsou souřadnice bodu  $A$  v TSS 2. Je nutné si uvědomit, že tyto prvky jsou konstantní v čase. Vektor si lze představit jako úsečku spojující počátek TSS 2 a bod  $A$ . Tato úsečka spojuje dva body téhož tělesa a tudíž nemění svojí délku, pokud uvažujeme tuhé těleso.

Rozšířený polohový vektor  $r_{1A}$  popisuje polohu bodu  $A$  v TSS 1. Tento vektor je již v čase proměnný, pokud předpokládáme, že se těleso 2 pohybuje vůči tělesu 1. Opět platí, že si lze představit tento vektor jako spojnicí počátku TSS1 a bodu  $A$ .

Matice  $T_{12}$  se nazývá transformační matice a je složena ze směrových kosinů mezi souřadnými osami systémů 1 a 2. Například prvek  $\cos \langle x_1, y_2 \rangle$  označuje směrový cosinus úhlu mezi osami  $x_1$  a  $y_2$ . Další prvky transformační matice jsou  $x_{02}, y_{02}, z_{02}$  a tyto označují souřadnice počátku TSS 2 v TSS 1. Na základě znalosti prvků této matice lze říci, že transformační matice závisí na vzájemné poloze počátků TSS a úhlech mezi jejich osami. Lze tedy říci že transformační matice charakterizuje spojení těles 1 a 2 neboli jejich vazbu. Matice je v čase proměnná; například pokud jsou tělesa spojena rotační vazbou, bude transformační matice bude funkcí úhlu natočení této vazby. Pro další využití je potřeba zavést submatici transformační matice  $T_{12}$ . Jedná se o matici směrových kosinů, která bude značena  $S_{12}$

$$S_{12} = \text{submatice}[T_{12}]^{1-3 \times 1-3}, \text{ takže } S_{12} = \begin{bmatrix} \cos \langle x_1, x_2 \rangle & \cos \langle x_1, y_2 \rangle & \cos \langle x_1, z_2 \rangle \\ \cos \langle y_1, x_2 \rangle & \cos \langle y_1, y_2 \rangle & \cos \langle y_1, z_2 \rangle \\ \cos \langle z_1, x_2 \rangle & \cos \langle z_1, y_2 \rangle & \cos \langle z_1, z_2 \rangle \end{bmatrix}. \quad (3.13)$$

Matice směrových kosinů  $S_{12}$  je ortogonální, což znamená, že platí

$$S_{12}^{-I} = S_{12}^T. \quad (3.14)$$

Pozn.: Pro submatici tvořenou z matice  $A$  bude v textu užíván následující zápis

$$A = \text{submatice}[B]^{1-3 \times 2-4}, \quad (3.15)$$

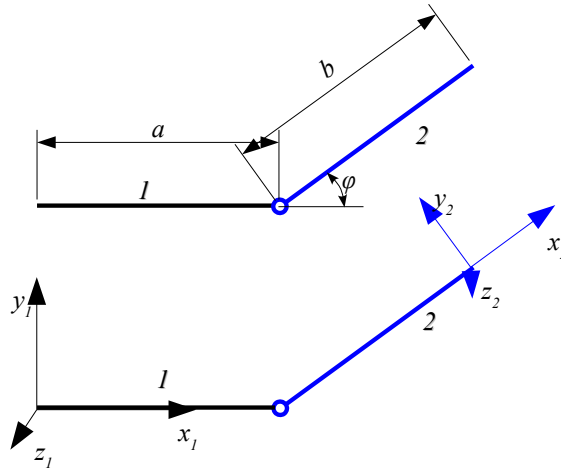
který znamená, že matice  $A$  je submaticí matice  $B$  a to tak, že jsou vybrány řádky 1 až 3 a sloupce 2 až 4.

### 3.3.2 Sestavování transformační matice pomocí matic základních pohybů

V technické praxi se transformační matice, která definuje spojení dvou těles, sestavuje pomocí součinu transformačních matic základních pohybů. Základními pohyby rozumíme tři posuvy ve směru souřadných os  $x, y, z$  a tři rotace kolem těchto os.

Například uvažujme dvě přímá tělesa o délkách  $a$  a  $b$ , která jsou spojena rotační vazbou, viz obr. 3.11. Transformační matici, která definuje přechod z TSS 1 do TSS 2, získáme tak, že si

představím pohyby, které jsou nezbytné pro přemístění TSS 1 do TSS 2. Nejprve posuneme souřadný systém ve směru osy  $x$  do vazby o vzdálenost  $a$ . Poté natočíme souřadný systém kolem osy  $z$  o souřadnici rotační vazby o  $\varphi$ . A konečně posune souřadný systém ve směru osy  $x$  o vzdálenost  $b$  tak, abychom se přemístili do polohy TSS 2. Takže transformační matici přechodu z TSS 1 do TSS 2 lze sestavit jako součin třech transformačních matic základních pohybů; konkrétně jako součin matice posuvu ve směru osy  $x$ , matice rotace kolem osy  $z$  a matice posuvu ve směru  $x$ .



Obrázek 3.11: Dvě přímá tělesa o délkách  $a$  a  $b$  spojena rotační vazbou

Definujme jednotlivé transformační matice základních pohybů, které budou označovány pro posuvné neboli translační pohyby jako  $T_x(a)$ ,  $T_y(b)$ ,  $T_z(c)$ , kde dolní index značí osu, po které se translace děje, a v závorce je velikost posuvu. Transformační matice pro rotační pohyby budou dále označovány jako  $R_x(\xi)$ ,  $R_y(\eta)$ ,  $R_z(\zeta)$ , kde dolní index značí osu rotace a v závorce je úhel pootočení. Z důvodu zjednodušení lze požit transformační matici posuvu  $T(A)$  do obecného bodu  $A$ . Tvary jednotlivých matic jsou následující

$$\begin{aligned}
 T_x(a) &= \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & T_y(b) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & T_z(c) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 T(A) &= \begin{bmatrix} 1 & 0 & 0 & x_A \\ 0 & 1 & 0 & y_A \\ 0 & 0 & 1 & z_A \\ 0 & 0 & 0 & 1 \end{bmatrix}, & R_x(\xi) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\xi) & -\sin(\xi) & 0 \\ 0 & \sin(\xi) & \cos(\xi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & (3.16) \\
 R_y(\eta) &= \begin{bmatrix} \cos(\eta) & 0 & \sin(\eta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\eta) & 0 & \cos(\eta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \text{a} & R_z(\zeta) &= \begin{bmatrix} \cos(\zeta) & -\sin(\zeta) & 0 & 0 \\ \sin(\zeta) & \cos(\zeta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
 \end{aligned}$$

Procedury napsané v programovacím jazyku používaném v softwaru MATLAB, které sestaví

transformační matici základního pohybu, jsou v příloze 11.1.

Nyní se vraťme k příkladu z obr. 3.11. Základní matice by pro tento příklad dvou těles spojených rotační vazbou byla získána pomocí základních pohybů následovně

$$\mathbf{T}_{12} = \mathbf{T}_x(a) \cdot \mathbf{R}_z(\varphi) \cdot \mathbf{T}_x(b). \quad (3.17)$$

O základních pohybech je třeba ještě poznamenat, že jejich složením lze realizovat jakýkoli pohyb. Dále platí, že pořadí posuvných pohybů je záměnné a také pořadí posuvného a rotačního pohybu ve směru a kolem stejné souřadné osy je záměnné.

### 3.3.3 Inverzní pohyb

Inverzním pohybem rozumíme pohyb „opačný“ vzhledem k původnímu pohybu tak, že posloupnost pohybů a jejich velikosti jsou opačné vzhledem k původnímu pohybu. V případě obr. 3.10 je pohyb neboli přechod definován z TSS 1 do TSS 2 a k tomuto lze přiřadit inverzní pohyb jako přechod z TSS 2 do TSS 1. Transformační matici inverzního pohybu lze získat dvěma způsoby, a to inverzí původní transformační matice

$$\mathbf{T}_{21} = (\mathbf{T}_{12})^{-1} \quad (3.18)$$

nebo součinem základních pohybů v opačném pořadí a s opačnými znaménky velikostí posunů a úhlů. Takže pro příklad rotační kinematické dvojice z obr. 3.11, jejíž transformační matice je 3.17, bude transformační matice inverzního pohybu mít následující tvar

$$\mathbf{T}_{21} = \mathbf{T}_x(-b) \cdot \mathbf{R}_z(-\varphi) \cdot \mathbf{T}_x(-a). \quad (3.19)$$

### 3.3.4 Rychlost bodu a úhlové rychlosti tělesa

Vyjděme z obr. 3.10 a pokusme se sestavit vztah pro rychlost bodu A tělesa 2 v TSS 1. Jinými slovy složky vektoru rychlosti budou vyjádřeny v TSS 1. Vyjdeme z rovnice 3.11, kterou derivujeme dle času, čímž získáme rozšířený vektor rychlosti bodu A v TSS 1

$$\mathbf{v}_{1A} = \dot{\mathbf{T}}_{12} \cdot \mathbf{r}_{2A}, \text{ kde } \mathbf{v}_{1A} = \begin{bmatrix} v_{1Ax} \\ v_{1Ay} \\ v_{1Az} \\ 0 \end{bmatrix}, \quad (3.20)$$

kde  $\dot{\mathbf{T}}_{12}$  je časová derivace transformační matice, kterou lze dále rozepsat jako

4 Vzhledem k malým rozměrům transformačních matic a výkonu výpočetní techniky je pohodlnější využít jednoduše inverzi původní matice.

$$\dot{\mathbf{T}}_{12} = \mathbf{T}_{12} \cdot \mathbf{V}_{12}, \quad (3.21)$$

kde  $\mathbf{V}_{12} \in \mathbb{R}^{4 \times 4}$  se nazývá matice rychlosti pohybu.<sup>5</sup> Matice rychlosti se sestavuje pomocí základních pohybů, viz kap. 3.3.6. Po dosazení 3.21 do 3.20 získáme vztah pro rychlost vyjádřený maticí rychlosti

$$\mathbf{v}_{1A} = \mathbf{T}_{12} \cdot \mathbf{V}_{12} \cdot \mathbf{r}_{2A}. \quad (3.22)$$

Pokud bychom potřebovali matici rychlosti inverzního pohybu zavedeného v kap. 3.3.3, tak pro matici rychlosti inverzního pohybu platí

$$\mathbf{V}_{21} = -\mathbf{T}_{12} \cdot \mathbf{V}_{12} \cdot \mathbf{T}_{12}^{-1}. \quad (3.23)$$

Z matice  $\mathbf{V}_{12}$  lze získat úhlové rychlosti tělesa 2 vůči tělesu 1, což je nezbytná veličina pro dynamiku těles. Matici úhlových rychlostí  $\boldsymbol{\Omega}_{12,2} \in \mathbb{R}^{3 \times 3}$  získáme jako submatici matice rychlosti  $\mathbf{V}_{12}$

$$\boldsymbol{\Omega}_{12,2} = \text{submatice}[\mathbf{V}_{12}]^{1-3 \times 1-3} \text{ nebo } \mathbf{V}_{12} = \begin{bmatrix} & & & \dots \\ & \boldsymbol{\Omega}_{12,2} & & \dots \\ & & & \dots \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (3.24)$$

přičemž byl využit zápis zavedený v 3.15. Dolní index za čárkou značí, že složky úhlových rychlostí jsou v TSS 2. Uspořádání úhlových rychlostí v matici  $\boldsymbol{\Omega}_{12,2}$  je následující

$$\boldsymbol{\Omega}_{12,2} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \quad (3.25)$$

Matici lze přepsat do vektoru

$$\boldsymbol{\omega}_{12,2} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}. \quad (3.26)$$

Dolní index za čárkou značí, že složky úhlových rychlostí jsou stále v TSS 2. Pokud bychom chtěli úhlové rychlosti převést do TSS 1, musíme transformovat matici směrových kosinů dle

$$\boldsymbol{\Omega}_{12,1} = \mathbf{S}_{12} \cdot \boldsymbol{\Omega}_{12,2} \text{ nebo } \boldsymbol{\omega}_{12,1} = \mathbf{S}_{12} \cdot \boldsymbol{\omega}_{12,2}. \quad (3.27)$$

<sup>5</sup> Zápisem  $\mathbf{A} \in \mathbb{R}^{m \times n}$  rozuměj, že  $\mathbf{A}$  je matice o  $m$  řádcích a  $n$  sloupcích.

### 3.3.5 Zrychlení bodu a úhlové zrychlení tělesa

Obdobně jako v předchozí kapitole nyní ukážeme, jak vyjádřit zrychlení bodu  $A$  z obr. 3.10 ve složkách TSS 1. Rozšířený vektor zrychlení bodu  $A$  získáme časovou derivací vztahu 3.20

$$\mathbf{a}_{1A} = \ddot{\mathbf{T}}_{12} \cdot \mathbf{r}_{2A} \quad (3.28)$$

a při využití vztahu 3.21 lze psát

$$\ddot{\mathbf{T}}_{12} = \dot{\mathbf{T}}_{12} \cdot (\mathbf{A}_{12} + \mathbf{V}_{12}^2), \text{ kde } \mathbf{A}_{12} = \dot{\mathbf{V}}_{12}. \quad (3.29)$$

Matice  $\mathbf{A}_{12} \in \mathbb{R}^{4 \times 4}$  se nazývá matice zrychlení pohybu. Sestavení matice zrychlení použitím základních pohybů je shrnut v následující kap. 3.3.6.

Matici zrychlení inverzního pohybu zavedeného v kap. 3.3.3 získáme z

$$\mathbf{A}_{21} = -\mathbf{T}_{12} \cdot \mathbf{A}_{12} \cdot \mathbf{T}_{12}^{-1}. \quad (3.30)$$

Z matice  $\mathbf{A}_{12}$  lze opět získat úhlové veličiny. Matici úhlových zrychlení  $\dot{\boldsymbol{\Omega}}_{12,2} \in \mathbb{R}^{3 \times 3}$  tělesa 2 vůči tělesu 1 získáme jako submatici matice zrychlení  $\mathbf{A}_{12}$

$$\dot{\boldsymbol{\Omega}}_{12,2} = \text{submatice} [\mathbf{A}_{12}]^{1-3 \times 1-3} \text{ nebo } \mathbf{A}_{12} = \begin{bmatrix} & \dot{\boldsymbol{\Omega}}_{12,2} & \cdots \\ 0 & 0 & 0 & 0 \end{bmatrix},^6 \quad (3.31)$$

přičemž byl využit zápis zavedený v 3.15. Dolní index za čárkou značí, že složky úhlových rychlostí jsou v TSS 2. Uspořádání úhlových zrychlení následující

$$\dot{\boldsymbol{\Omega}}_{12,2} = \begin{bmatrix} 0 & -\alpha_z & \alpha_y \\ \alpha_z & 0 & -\alpha_x \\ -\alpha_y & \alpha_x & 0 \end{bmatrix}. \quad (3.32)$$

Matici lze přepsat do vektoru

$$\boldsymbol{\alpha}_{12,2} = \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{bmatrix}. \quad (3.33)$$

Pokud bychom chtěli úhlové zrychlení převést do TSS 1, musíme transformovat maticí směrových kosinů dle

$$\dot{\boldsymbol{\Omega}}_{12,1} = \mathbf{S}_{12} \cdot \dot{\boldsymbol{\Omega}}_{12,2} \text{ nebo } \boldsymbol{\alpha}_{12,1} = \mathbf{S}_{12} \cdot \boldsymbol{\alpha}_{12,2}. \quad (3.34)$$

6 Jako u úhlových rychlostí, by bylo logické označit matici úhlových zrychlení velkým písmenem řecké abecedy, takže  $\dot{\boldsymbol{\Omega}}_{12,2}$ . Podoba velkého řeckého písmene  $\mathbf{A}$  a velkého latinského písmene  $\mathbf{A}$  označení by mohla vést čtenáře k záměně. Proto jsme využili zápis  $\dot{\boldsymbol{\Omega}}_{12,2}$ .

### 3.3.6 Matice zrychlení a rychlosti základních pohybů

V kap. 3.3.4 a 3.3.5 byly uvedeny matice rychlosti  $V_{I2}$  a zrychlení  $A_{I2}$ . Tyto matice se v praxi stejně jako transformační matice sestavují pomocí základních pohybů definovaných v kap. 3.3.2.

Nejprve definujeme tzv. maticové diferenciální operátory. Každému ze šesti základních pohybů odpovídá jeden maticový diferenciální operátor, např. operátor  $D_{Tx}$  odpovídá posuvu ve směru osy  $x$  nebo operátor  $D_{Rx}$  přísluší rotaci kolem osy  $x$ . Diferenciální operátory základních pohybů mají následující tvar

$$\begin{aligned} D_{Tx} &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & D_{Ty} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & D_{Tz} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\ D_{Rx} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & D_{Ry} &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \text{a } D_{Rz} &= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \end{aligned} \quad (3.35)$$

Matice rychlosti respektive zrychlení pro základní pohyby lze získat jednoduše vynásobením diferenciálního operátoru rychlostí respektive zrychlením souřadnice vazby

$$V_i = D_i \cdot \dot{q}_i, \text{ respektive } A_i = D_i \cdot \ddot{q}_i, \quad (3.36)$$

kde  $i$  označuje kterýkoli základní pohyb. Uvažujme například dvě tělesa spojená rotační vazbou z obr. 3.11. Transformační matice přechodu z tělesa 1 do tělesa 2 je 3.17. Jedinou časově proměnnou složkou této transformační matice je souřadnice rotační vazby  $\varphi$ . Takže matice rychlosti by měla následující tvar

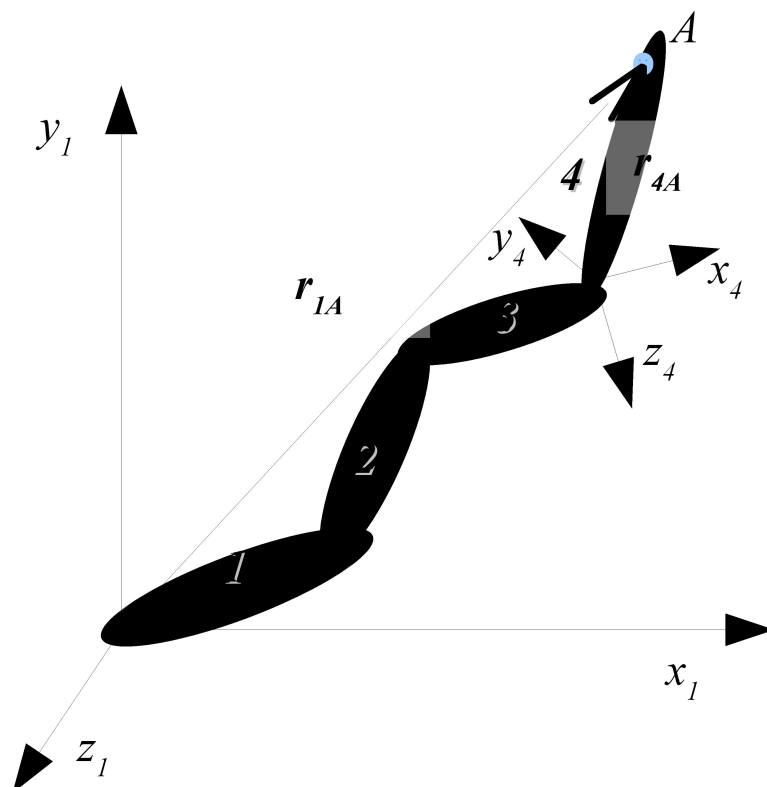
$$V_{I2} = D_{Rz} \cdot \dot{\varphi} = \begin{bmatrix} 0 & -\dot{\varphi} & 0 & 0 \\ \dot{\varphi} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.37)$$

a matice zrychlení by byla

$$A_{I2} = D_{Rz} \cdot \ddot{\varphi} = \begin{bmatrix} 0 & -\ddot{\varphi} & 0 & 0 \\ \ddot{\varphi} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (3.38)$$



## 3.3.7 Spojení více těles



Obrázek 3.12: Spojení čtyř těles do jednoduchého řetězce

Nyní se budeme zabývat popisem pohybu více než dvou těles spojených do jednoduchého kinematického řetězce – jako příklad je uvedeno spojení čtyř těles, která jsou spojena libovolnými vazbami, viz obr. 3.12. Cílem je nalézt matici  $T_{14}$ , která definuje přechod od tělesa 1 až po těleso 4. Matice  $T_{14}$  zahrnuje popis vazeb a poloh vazeb všech přítomných těles. Pokud by například mezi tělesem 3 a 4 byla kloubová vazba, tak matice  $T_{14}$  bude mimo jiné také funkcí třech úhlů natočení tohoto kloubu. Matici  $T_{14}$  získáme podobným principem jaký jsme užili u základních pohybů, což je postupný součin transformačních matic, které definují spojení jednotlivých kinematických dvojic

$$T_{14} = T_{12} \cdot T_{23} \cdot T_{34}. \quad (3.39)$$

Tento vztah lze zobecnit pro  $n$  těles následovně

$$T_{1n} = T_{1,2} \cdot T_{2,3} \cdot \dots \cdot T_{n-1,n}. \quad (3.40)$$

Pokud bychom chtěli vyšetřovat trajektorii bodu  $A$  v TSS 1, tak lze využít vztahu

$$r_{1A} = T_{1n} \cdot r_{nA}. \quad (3.41)$$

### 3.3.8 Úloha polohy maticovou metodou

Jak již bylo popsáno v kap. 3.2.5, úloha polohy slouží ke stanovení souřadnic vazeb, ze kterých lze určit polohy jednotlivých členů mechanismu a hodnoty výstupních souřadnic v závislosti na vstupních souřadnicích. Nyní popíšeme postup získání soustavy rovnic definované v 3.4, která popisuje polohu mechanismu.

#### Úloha polohy pro mechanismus s jednou základní smyčkou

Po rozboru uspořádání mechanismu a jeho počtu základních smyček dle kap. 3.2.4 lze říci, že dle 3.3 pro mechanismus s jednou základní smyčkou

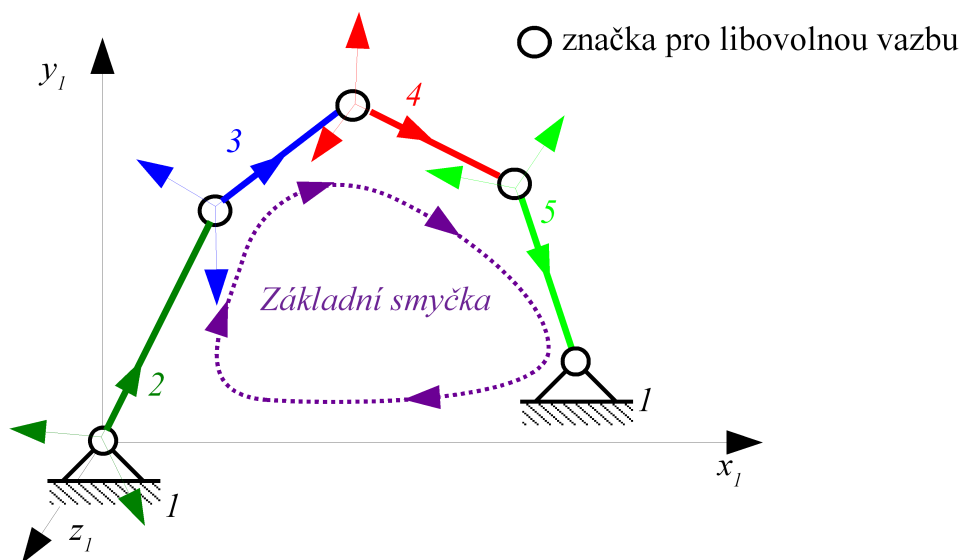
$$r = 1 \quad (3.42)$$

získáme soustavu

$$r \cdot 6 = 6 \quad (3.43)$$

rovníc.

Uvažujme mechanismus, jehož základní smyčka je zjednodušeně zobrazena na obr. 3.13. Předpokládejme, že jsme již sestavili kinematické schéma, definovali rozměry mechanismu, stanovili počet stupňů volnosti soustavy  $i$  a vstupní souřadnice  $s$  a určili jsme, které výstupní souřadnice nebo jiné výstupní kinematické veličiny chceme zjistit.



Obrázek 3.13: Zjednodušené schéma mechanismu s jednou základní smyčkou

Všem členům přiřadíme tělesové souřadné systémy s příslušnými pořadovými čísly (připomeňme, že všechny systémy musí být pravotočivé) jako v obr. 3.13. Nyní sestavíme posloupnost členů tak, jak jsou seřazeny v základní smyčce. V tomto případě je posloupnost členů

ve smyčce následující: 1, 2, 3, 4, 5 a 1. Pomocí transformačních rovnic se budeme postupně přesouvat celým řetězcem. Vycházíme z rámu 1. Pomocí transformační matice  $T_{12}$  se přemístíme do prostoru členu 2 a tak dále, až se pomocí matice  $T_{51}$  vrátíme zpět do prostoru rámu 1. Nyní již stačí matematicky zapsat maticovou rovnici základní smyčky jako

$$\mathbf{E} = \mathbf{T}_{12} \cdot \mathbf{T}_{23} \cdot \mathbf{T}_{34} \cdot \mathbf{T}_{45} \cdot \mathbf{T}_{51}, \quad (3.44)$$

kde  $\mathbf{E} \in \mathbb{R}^{4 \times 4}$  je matice identity neboli jednotková matice,  $T_{12}$  je transformační matice přechodu z rámu 1 do tělesa 2 apod. Transformační matice jsou sestaveny skládáním transformačních matic základních pohybů, viz. kap. 3.3.2. Vztah 3.44 je maticovou rovnicí mechanismu, která představuje šestnáct skalárních a nelineárních rovnic, z nichž dle předchozího popisu pouze šest je nezávislých a využitelných. Z maticové rovnice 3.44 lze vybrat šest skalárních rovnic například podle následujícího schématu

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \dots & f_1(s, l, q) & f_2(s, l, q) & f_3(s, l, q) \\ \dots & \dots & f_4(s, l, q) & f_5(s, l, q) \\ \dots & \dots & \dots & f_6(s, l, q) \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3.45)$$

kde  $s$  označuje vektor vstupních souřadnic,  $l$  vektor rozměrů mechanismu a  $q$  je neznámý vektor souřadnic vazeb (úhly natočení nebo posunutí). Přepis soustavy rovnic je běžné podoby je

$$\begin{aligned} f_1(s, l, q) &= 0, \\ f_2(s, l, q) &= 0, \\ f_3(s, l, q) &= 0, \\ f_4(s, l, q) &= 0, \\ f_5(s, l, q) &= 0, \\ f_6(s, l, q) &= 0. \end{aligned} \quad (3.46)$$

Tímto získáváme rovnice již uvedené a popsané ve vztahu 3.4. Pokud vyřešíme tuto soustavu šesti nelineárních, transcendentních a skalárních rovnic, máme polohu mechanismu zcela popsanou.

### Úloha polohy pro mechanismus s $r$ základními smyčkami

Mechanismus tvořený  $r$  základními smyčkami se řeší obdobně jak je popsáno v předešlé podkapitole. Popsaný postup jednoduše  $r$  krát zopakujeme, takže tvoříme  $r$  maticových rovnic jako 3.46.

Jako příklad uvažujme mechanismus složený ze dvou základních smyček

$$r = 2, \quad (3.47)$$

jehož zjednodušené schéma je na obr. 3.7. Dle 3.3 bude mechanismus popsán soustavou

$$r \cdot 6 = 12 \quad (3.48)$$

dvanácti rovnic. Základní smyčky jsou definovány následně:

- Základní smyčka I: 1, 2, 3, 4, 1.
- Základní smyčka II: 1, 7, 6, 5, 3, 4, 1.

Pro takto uspořádané smyčky by maticové rovnice definované v 3.44 nabýly následujícího tvaru

$$\begin{aligned} \text{Základní smyčka I: } \mathbf{E} &= \mathbf{T}_{12} \cdot \mathbf{T}_{23} \cdot \mathbf{T}_{34} \cdot \mathbf{T}_{41}, \\ \text{Základní smyčka II: } \mathbf{E} &= \mathbf{T}_{17} \cdot \mathbf{T}_{76} \cdot \mathbf{T}_{65} \cdot \mathbf{T}_{53} \cdot \mathbf{T}_{34} \cdot \mathbf{T}_{41}, \end{aligned} \quad (3.49)$$

Jedná se o soustavu třiceti dvou skalárních rovnic, z nichž lze využít dle 3.48 pouze dvanáct rovnic použitím schématu uvedeném v 3.45. Pro přehlednost jsou tyto nezávislé skalární rovnice zvýrazněny

$$\begin{aligned} \text{Základní smyčka I: } \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} &= \begin{bmatrix} \dots & f_1(s, l, q) & f_2(s, l, q) & f_3(s, l, q) \\ \dots & \dots & f_4(s, l, q) & f_5(s, l, q) \\ \dots & \dots & \dots & f_6(s, l, q) \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ \text{Základní smyčka II: } \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} &= \begin{bmatrix} \dots & f_7(s, l, q) & f_8(s, l, q) & f_9(s, l, q) \\ \dots & \dots & f_{10}(s, l, q) & f_{11}(s, l, q) \\ \dots & \dots & \dots & f_{12}(s, l, q) \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (3.50)$$

Získáváme soustavu dvanácti nelineárních, transcendentních a skalárních rovnic, která zcela popisuje polohu celého mechanismu z obr. 3.44. Neznámými v těchto rovnicích jsou souřadnice všech vazeb  $q$ . Praktické příklady mechanismů a sestavení jejich úloh polohy jsou uvedeny v kap. 6.2, 7.6, 8.4 a 8.9.3.

### 3.3.9 Rychlosti a zrychlení mechanismů

Hned úvodem upozorňujeme, že tvorba rovnic rychlostí a zrychlení je značně komplikovaná a proto v následující kap. 3.3.10 navrhuje jednodušší způsob, jak lze zjistit rychlosti a zrychlení numericky.

Rychlosti mechanismu stanovíme derivací úlohy polohy definované v předchozí kap. 3.3.8. Například pro mechanismus s jednou základní smyčkou, který je vyobrazen na obr. 3.13 a popsán úlohou polohy 3.44 by maticová rovnice pro rychlosti mechanismu byla následující

$$\mathbf{0} = \mathbf{T}_{12} \cdot \mathbf{V}_{12} \cdot \mathbf{T}_{23} \cdot \mathbf{T}_{34} \cdot \mathbf{T}_{45} \cdot \mathbf{T}_{51} + \mathbf{T}_{12} \cdot \mathbf{T}_{23} \cdot \mathbf{V}_{23} \cdot \mathbf{T}_{34} \cdot \mathbf{T}_{45} \cdot \mathbf{T}_{51} + \dots + \mathbf{T}_{12} \cdot \mathbf{T}_{23} \cdot \mathbf{T}_{34} \cdot \mathbf{T}_{45} \cdot \mathbf{T}_{51} \cdot \mathbf{V}_{51}. \quad (3.51)$$

Rovnice není rozepsána celá. Rozepsaná rovnice by měla 5 sčítanců. Tato rovnice představuje

maticovou, lineární rovnici, ze které lze použít opět pouze 6 skalárních rovnic dle vzoru uvedeném v 3.45. Tyto rovnice představují soustavu 6 lineárních rovnic, kde neznámými jsou rychlosti vazbových zobecněných souřadnic (posunutí nebo natočení).

Pokud potřebujeme získat rychlost konkrétního bodu, použijeme vztah 3.22, do kterého dosadíme vypočtené rychlosti vazbových souřadnic vypočtené z 3.51. Obdobně postupujeme při výpočtu úhlových rychlostí, kdy využijeme vztahu 3.24.

Pokud bychom uvažovali mechanismus s  $r$  základními smyčkami, získali bychom  $r$  maticových rovnic typu 3.51. Takže například pro mechanismus na obr. 3.7 s 2 základními smyčkami bychom derivovali rovnice 3.49 podle vzoru 3.51.

Při výpočtu zrychlení mechanismu postupujeme obdobně jako při výpočtu rychlostí. Uveďme opět provedení pro mechanismus s jednou základní smyčkou z obr. 3.13, jehož rychlosti jsou popsány v 3.51. Časová derivace této rovnice vyjadřující zrychlení by měla následující tvar

$$\begin{aligned} \mathbf{0} = & T_{12} \cdot A_{12} \cdot T_{23} \cdot T_{34} \cdot T_{45} \cdot T_{51} + T_{12} \cdot V_{12}^2 \cdot T_{23} \cdot T_{34} \cdot T_{45} \cdot T_{51} + T_{12} \cdot V_{12} \cdot T_{23} \cdot V_{23} \cdot T_{34} \cdot T_{45} \cdot T_{51} + \\ & + T_{12} \cdot V_{12} \cdot T_{23} \cdot T_{34} \cdot V_{34} \cdot T_{45} \cdot T_{51} + T_{12} \cdot V_{12} \cdot T_{23} \cdot T_{34} \cdot T_{45} \cdot V_{45} \cdot T_{51} + \\ & + T_{12} \cdot V_{12} \cdot T_{23} \cdot T_{34} \cdot T_{45} \cdot T_{51} \cdot V_{51} + \dots \end{aligned} \quad (3.52)$$

Pro tuto rovnici platí stejná pravidla jako byla uvedena pro rovnici 3.51 s jedinou výjimkou, a to že výstupem jsou zrychlení souřadnic vazeb. Všimněme si, že rovnice není kompletní. Celá rozepsaná rovnice by měla 30 sčítanců. Vzhledem k rozsáhlosti této rovnice stoupá pravděpodobnost, že se řešitel při jejich sestavování dopustí chyby, a to zvláště pokud mechanismus má více než jednu základní smyčku a řešitel musí sestavovat rovnice jako je 3.51 a 3.52 tolikrát, kolik je základních smyček. Proto navrhuje stanovovat zrychlení a rychlosti pomocí numerické derivace, viz následující kapitola.

### 3.3.10 Numerické určování rychlostí a zrychlení mechanismu

Tabulka 3.3: Koeficienty pro výpočet numerické derivace

		Pořadová čísla bodů, kterými se určuje derivace					
Řád derivace	Řád přesnosti	0	1	2	3	4	5
1	1	-1	1				
	2	-3/2	2	-1/2			
	3	-11/6	3	-3/2	1/3		
	4	-25/12	4	-3	4/3	-1/4	
2	1	1	-2	1			
	2	2	-5	4	-1		
	3	35/12	-26/3	19/2	-14/3	11/12	
	4	15/4	-77/6	107/6	-13	61/12	-5/6

Nejprve uveďme postup pro výpočet numerické derivace funkce. K výpočtu derivace slouží tabulka 3.3, která uvádí hodnoty koeficientů, kterými se násobí hodnoty funkcí v různých bodech. Použití tabulky lze nejlépe vysvětlit na příkladu. Uvažujme funkci  $f(x)$ , u které chceme vyčíslit hodnotu její druhé derivace v bodě  $x_0$ , a to s přesností 2. řádu. Numerická hodnota derivace bude zjištěna vztahem

$$f''(x_0) = \frac{2 \cdot f(x_0 - h) - 5 \cdot f(x_0) + 4 \cdot f(x_0 + h) - 1 \cdot f(x_0 + 2 \cdot h)}{h^2} + O(h^2), \quad (3.53)$$

kde  $h$  je krok nebo „vzdálenost“, na které určujeme hodnotu numerické derivace.  $O(h^2)$  je chybová funkce numerické derivace, která je řádu  $h^2$ . Z tabulky vyplývá, že čím vyšší derivaci potřebujeme a čím menší chybu neboli vyšší řád přesnosti vyžadujeme, v o to větším počtu bodů musíme funkci vyčíslit. Doporučení pro volbu velikosti kroku  $h$  je uvedeno na konci této podkapitoly.

Na základě výkladu o numerické derivaci lze numericky určovat matice rychlosti i zrychlení. Mějme pohyb, který je definován transformační maticí  $T_{12}$ . Vzhledem k tomu, že uvažujeme mechanismus, který se pohybuje, je matice  $T_{12}$  závislá na čase. Uvažujme-li nejnižší řád přesnosti, tak první časová derivace matice  $T_{12}$  v okamžiku  $t_0$  je dána vztahem

$$\dot{T}_{12}(t_0) = (-T_{12}(t_0) + T_{12}(t_0 + h)) \cdot \frac{1}{h}, \quad (3.54)$$

kde  $h$  je v tomto případě časový krok, na kterém je určována numerická derivace. Matici  $\dot{T}_{12}(t_0)$  lze přímo využít dle vztahu 3.20 pro výpočet rychlosti bodu. Je nutné upozornit, že matice

$\dot{T}_{12}(t_0)$  je platná pouze pro okamžik  $t_0$ , v jiném čase musíme opět vyčíslit vztah 3.54. Při výpočtu

úhlových rychlostí tělesa 2 v TSS 1 bychom potřebovali matici rychlosti pohybu  $V_{12}$ , kterou pak lze stanovit využitím 3.21 jako

$$V_{12}(t_0) = T_{12}(t_0)^{-1} \cdot \dot{T}_{12}(t_0). \quad (3.55)$$

Úhlové rychlosti poté stanovíme dle postupu uvedeného v kap. 3.3.4. Samozřejmě lze užít vztahy s vyššími řády přesnosti dle tab. 3.3.

Pro výpočet zrychlení platí podobné úvahy jako pro rychlosti. Druhou časovou derivaci matice  $T_{12}$  v čase  $t_0$  vypočteme s přesností prvního řádu dle vztahu

$$\ddot{T}_{12}(t_0) = (T_{12}(t_0 - h) - 2 \cdot T_{12}(t_0) + T_{12}(t_0 + h)) \cdot \frac{1}{h^2}. \quad (3.56)$$

Tuto matici lze přímo využít při výpočtu zrychlení bodu dle 3.28. Pro výpočet matice zrychlení  $A_{12}(t_0)$  lze využít 3.30

$$A_{12}(t_0) = T_{12}(t_0)^{-1} \cdot \ddot{T}_{12}(t_0) - V_{12}^2(t_0). \quad (3.57)$$

Další výklad o zrychlení tělesa je uveden v kap. 3.3.5.

### Využití převodových funkcí

Jak bylo popsáno v kap. 3.2.7, lze u mechanismů využít pro výpočet rychlostí a zrychlení tzv. převodové funkce a jejich derivace. Tohoto bude dále využito v návrhu výpočtu dynamiky mechanismu v kap. 6.

Uvažujme mechanismus s jednou souřadnicí mechanismu  $s_1$ . Potom pro rychlost kterékoli výstupní souřadnice mechanismu platí dle 3.7 vztah využívající převodu

$$\dot{q}_I = p_{I,1} \cdot \dot{s}_1, \text{ kde } p_{I,1} = \frac{d q_I}{d s_1} \quad (3.58)$$

a zrychlení této výstupní souřadnice je dle 3.9

$$\ddot{q}_I = p_{I,1} \cdot \ddot{s}_1 + dp_{I,1} \cdot \dot{s}_1^2. \quad (3.59)$$

Tyto dva vztahy 3.58 a 3.59 lze využít pro jakoukoli polohovou veličinu mechanismu a tím pro ni vyčíslit rychlost a zrychlení. Například pro úhel natočení nebo posunutí vazby lze vypočítat odpovídající rychlost a zrychlení, viz kap. 6.3. Převod  $p_{I,1}$  a jeho derivaci  $dp_{I,1}$  lze vypočítat taktéž numerickou derivací.

Obdobný vztah lze zavést pro časovou derivaci transformační matice

$$\dot{T}_{12} = \frac{d T_{12}}{d t} = \frac{d T_{12}}{d t} \cdot \frac{d s_1}{d s_1} = \frac{d T_{12}}{d s_1} \cdot \dot{s}_1 = P_{12,1} \cdot \dot{s}_1, \quad (3.60)$$

Všimneme-li si podobnosti vztahů 3.58 a 3.60, lze říci, že derivaci  $T_{12}$  podle souřadnice mechanismu lze nazývat převod. Zavedme označení pro převod transformační matice

$$P_{12,I} = \frac{dT_{12}}{ds_I}, \quad (3.61)$$

kde  $P_{12,I} \in \mathbb{R}^{4 \times 4}$  je matice stejných rozměrů jako transformační matice. Matice rychlosti pohybu s využitím 3.21 bude

$$V_{12} = T_{12}^{-I} \cdot P_{12,I} \cdot \dot{s}_I, \quad (3.62)$$

kde derivaci transformační matice  $T_{12}$  podle souřadnice mechanismu  $s_I$  lze opět získat numericky dle tab. 3.3. Při uvažování nejnižšího řádu bude derivace transformační matice

$$P_{12,I} = \frac{dT_{12}}{ds_I} = (-T_{12}(s_I) + T_{12}(s_I + h)) \cdot \frac{1}{h}, \quad (3.63)$$

kde tentokrát je  $h$  krok vstupní souřadnice  $s_I$ , na kterém je určována numerická derivace.

Nyní odvodme vztahy pro zrychlení. Derivujme vztah 3.60 dle času

$$\ddot{T}_{12} = \frac{d}{dt}(P_{12,I} \cdot \dot{s}_I) = P_{12,I} \cdot \ddot{s}_I + \frac{dP_{12,I}}{ds_I} \cdot \dot{s}_I^2, \quad (3.64)$$

čímž získáváme analogický vztah k 3.58. Zavedme označení pro derivaci převodu

$$DP_{12,II} = \frac{dP_{12,I}}{ds_I}, \quad (3.65)$$

který lze opět stanovit numerickou derivací. Potom vztah 3.64 lze přepsat

$$\ddot{T}_{12} = P_{12,I} \cdot \ddot{s}_I + DP_{12,II} \cdot \dot{s}_I^2, \quad (3.66)$$

kde vystupuje převod transformační matice a jeho derivace. Matici zrychlení lze získat ze vztahu

$$A_{12} = T_{12}^{-I} \cdot \ddot{T}_{12} - V_{12}^2, \quad (3.67)$$

ze kterého po dosazení z 3.66 a 3.62 získáme

$$A_{12} = T_{12}^{-I} \left[ P_{12} \cdot \ddot{s}_I + (DP_{12} - P_{12} \cdot T_{12}^{-I} \cdot P_{12}) \dot{s}_I^2 \right]. \quad (3.68)$$

## Volba kroku h

Při volbě kroku  $h$  je třeba vzít v úvahu několik hledisek. Obecným pravidlem dle [15] je volit

$$h = \sqrt{u}, \quad (3.69)$$

kde  $u$  je přesnost použitého softwaru, která se pohybuje v řádech  $10^{-16}$ .



Pro numerické výpočty časových derivací respektive derivací podle vstupní souřadnice transformační matice (viz 3.54, 3.56, 3.63 apod) je nutno vyhodnotit úlohu polohy v několika časových okamžicích respektive pro několik hodnot vstupních souřadnic. Úloha polohy mechanismu je řešena některou z metod řešení soustav nelineárních rovnic uvedených v kap. 5. A právě zde je nutné si uvědomit, že přesnost výpočtu soustav nelineárních rovnic  $\varepsilon_{NS}$  musí být řádově mnohonásobně vyšší než zvolený krok  $h$ . Mohlo by se stát, že dovolená chyba výpočtu úlohy polohy  $\varepsilon_{NS}$  a současně matice  $T_{12}$  by byla srovnatelná s krokem  $h$ , na kterém stanovujeme numerickou derivaci, čímž bychom získali zcela chybné výsledky. Doporučení z autorovy praxe pro volbu  $h$  v závislosti na  $\varepsilon_{NS}$  je následující

$$h \geq 10^6 \cdot \varepsilon_{NS}. \quad (3.70)$$

Co se týče volby  $\varepsilon_{NS}$ , neměli bychom se zde řádově blížit přesnosti výpočetního stroje a použitého programového prostředí  $u$ , která se běžně pohybuje v řádu  $10^{-16}$ . Autorovo doporučení volby  $\varepsilon_{NS}$  je následující

$$\varepsilon_{NS} > 10^5 \cdot u. \quad (3.71)$$

## 4 Syntéza mechanismů

Syntézou mechanismů rozumíme postup, kterým hledáme rozměry nebo jiné parametry mechanismu tak, aby soustava plnila požadovanou funkci nebo měla žádané vlastnosti. Rozměry nebo parametry mechanismu, jejichž hodnoty budeme procesem syntézy hledat, budeme nazývat ladicí parametry.<sup>7</sup> Ladicí parametry budou řazeny do vektoru a značeny  $\mathbf{x}$ . Velikost vektoru  $\mathbf{x}$  a současně počet ladicích parametrů bude značen  $n$

$$\mathbf{x} \in \mathbb{R}^n. \quad (4.1)$$

Počet ladicích parametrů  $n$  budeme často nazývat rozměr úlohy syntézy nebo rozměr optimalizační úlohy.

V tomto textu se omezíme na syntézu pouze kinematických vlastností mechanismů.

Syntéza mechanismů lze rozdělit do těchto základních skupin:

1. Grafické metody – byly využívány dříve zvláště u rovinných mechanismů a dnes se již téměř nevyužívají, blíže viz [16].
2. Analytické metody – jedná se o skupinu metod, například metoda kolokační, kde hledáme pro  $n$  poloh mechanismu  $n$  ladicích parametrů.
3. Implementace optimalizačních metod – k syntéze mechanismů lze využít optimalizační metody, kdy lze optimalizovat jakoukoli vlastnost soustavy v libovolném počtu jejích poloh.

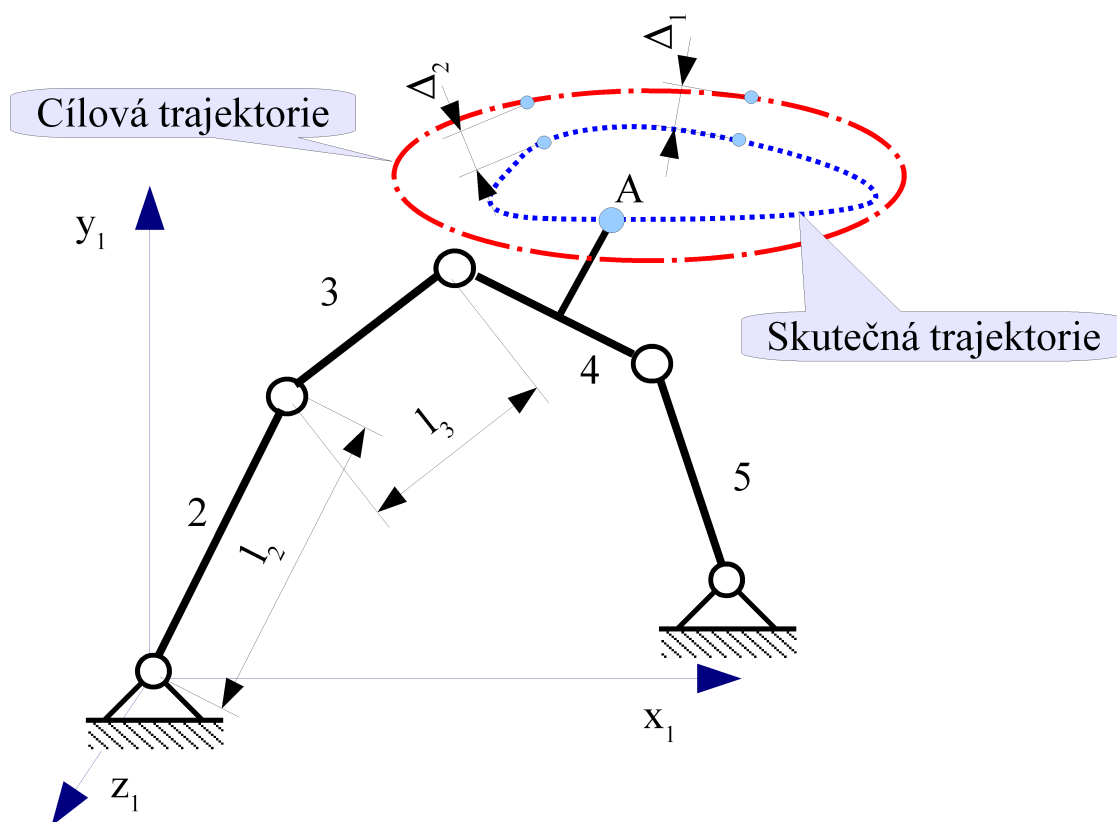
Analytické metody syntézy mechanismů jsou již detailně prozkoumány a popsány v mnoha zdrojích, viz [1], [2], [3], [4], [5] a [14]. Pro většinu technických problémů jsou tyto metody nedostatečné, protože jsou omezeny počtem zkoumaných poloh a jím odpovídajícím počtem ladicích parametrů. Například uvažujme vodicí mechanismus, jehož cílem je pohybovat se po určité trajektorii. U tohoto mechanismu bude možno změnit dva rozměry, takže lze zajistit kolokační metodou, že soustava bude přesně kopírovat danou trajektorii pouze ve dvou polohách. Může se stát, že v jiné poloze se bude skutečná trajektorie od žádané diametrálně lišit.

Naopak využití moderních optimalizačních metod k syntéze mechanismů umožňuje navrhnout jakýkoli počet ladicích parametrů tak, aby se pohybovaly co nejblíže žádané trajektorii, a to během celého pracovního cyklu. Zde je patrná výhoda využití optimalizačních, kde nejsme limitováni žádnými omezeními. Využití optimalizace k syntéze soustav není v literatuře dostatečná a proto se jí budeme nyní podrobně zabývat.

---

<sup>7</sup> Pro ladicí parametry je možno použít termínu proměnné návrhu nebo návrhové proměnné.

## 4.1 Formulace optimalizačního problému při syntéze mechanismů



Obrázek 4.1: ukázka vodícího mechanismu se skutečnou a cílovou trajektorií bodu  $A$

Nejprve popíšeme názorně formulaci optimalizačního problému na příkladu z obr. 4.1, kde je zobrazen mechanismus skládající se z pěti členů. Cílem syntézy je, aby se bod  $A$  členu 4 pohyboval po žádané neboli cílové trajektorii. Je dovoleno měnit délku členu 2 a 3 s tím, že člen 3 nesmí být delší než dovolená hodnota  $l_{3max}$ . Délky  $l_2$  a  $l_3$  jsou ladicími parametry soustavy, které uspořádáme do vektoru ladicích parametrů  $\mathbf{x}$

$$\mathbf{x} = \begin{bmatrix} l_2 \\ l_3 \end{bmatrix}, \quad (4.2)$$

takže rozměr optimalizační úlohy je

$$n=2. \quad (4.3)$$

Nyní, abychom mohli přejít k optimalizaci, je třeba stanovit funkci, která bude číselně vyjadřovat, zda se mechanismus pohybuje po žádané trajektorii spíše však jak blízko této trajektorie se pohybuje. Funkce bude dále nazývána cílová funkce a bude značena  $F$ .<sup>8</sup> Cílovou funkci stanovíme tak, že její minimum odpovídá optimu. Optimum bychom u úlohy z obr. 4.1 nazývali stav, kdyby bod

<sup>8</sup> Pro cílovou funkci je též vžitý název cenová funkce

A přesně kopíroval cílovou trajektorii vyznačenou červeně. Pro tento konkrétní případ musí být cílová funkce vypočtena z odchylek skutečné a cílové trajektorie a může být formulována jako

$$F(l_2, l_3) = F(\mathbf{x}) = \Delta_1^2 + \Delta_2^2 + \dots + \Delta_k^2, \quad (4.4)$$

kde  $k$  je počet poloh mechanismu, ve kterých provádíme výpočet odchylky skutečné od cílové trajektorie. Z definice cílové funkce vyplývá, že je nutno vyhodnotit úlohu polohy mechanismu v  $k$  polohách. Kvadráty jsou v cílové funkci z důvodu toho, že chceme minimalizovat i záporné odchylky. Nutno poznamenat, že z matematických důvodů jsou kvadráty pro optimalizaci vhodnější než absolutní hodnoty (vznik nehladkých a tudíž nediferencovatelných funkcí). Pro úplnost charakterizujeme cílovou funkci z hlediska vstupů a výstupů, a to

$$F(\mathbf{x}): \mathbb{R}^2 \rightarrow \mathbb{R}, \quad (4.5)$$

což znamená, že argumentem cílové funkce je reálný vektor ladicích parametrů se dvěma prvky a výsledkem funkce je reálná skalární hodnota.

Dále nesmíme zapomínat na to, že jsme limitováni při výběru hodnoty ladicího parametru  $l_3$  tak, že

$$l_3 \leq l_{3max} \rightarrow l_3 - l_{3max} \leq 0. \quad (4.6)$$

Tato limitace volby ladicího parametru se nazývá v optimalizaci omezovací podmínky nebo pouze omezení.

Nyní lze formulovat úlohu optimalizace pro uvedený příklad z obr. 4.1. Úlohou optimalizace je nalézt ladicí parametry  $l_2$  a  $l_3$  tak, aby hodnota cílové funkce  $F$  byla minimální a současně nesmí být porušena podmínka omezení 4.6.

## Obecná formulace úlohy optimalizace

Formulaci úlohy optimalizace zobecníme a definujeme matematicky – optimalizace je minimalizace cílové funkce prostřednictvím ladicích parametrů tak, abychom současně dodrželi podmínky omezení. Označíme-li vektor  $n$  ladicích parametrů jako  $\mathbf{x} \in \mathbb{R}^n$  a cílovou funkci jako  $F: \mathbb{R}^n \rightarrow \mathbb{R}$ , lze formulovat úlohu matematické optimalizace jako hledání optimálního vektoru ladicích parametrů, který označíme  $\mathbf{x}_{min}$ , což lze zapsat jako

$$\mathbf{x}_{min} : \min_{\mathbf{x} \in \mathbb{R}^n} F(\mathbf{x}) \quad \text{omezeno} \quad \begin{cases} h_i(\mathbf{x}) \leq 0, \text{ kde } i=1 \dots r \\ g_j(\mathbf{x}) = 0, \text{ kde } j=1 \dots s \end{cases}, \quad (4.7)$$

kde  $h_i(\mathbf{x})$  jsou nerovnostní podmínky omezení a  $g_j(\mathbf{x})$  jsou rovnostní podmínky omezení a  $r$  a  $s$  označují jejich počty.

Z hlediska matematiky je nezbytné úlohy optimalizace dělit na:

1. Optimalizace bez omezení – chybí omezovací podmínky  $h_i(\mathbf{x})$  a  $g_j(\mathbf{x})$ .
2. Optimalizace s omezením – existují omezovací podmínky  $h_i(\mathbf{x})$  a  $g_j(\mathbf{x})$ .

Algoritmy těchto dvou typů optimalizačních úloh se liší a proto je zcela nezbytné toto rozdělení zásadně dodržovat.

### **Charakteristika cílových funkcí a omezovacích podmínek v syntéze mechanismů**

Nyní si všimněme charakteru cílové funkce  $F$ , jak je definována v 4.4. Tento typ cílové funkce, kdy pracujeme s odchylkami trajektorií, se při syntéze mechanismů vyskytuje velice často. Dále je nezbytné si uvědomit, že pro jeden výpočet cílové funkce je potřeba vyhodnotit úlohu polohy mechanismu v několika polohách, což je další typický rys cílových funkcí mechanismů.

Cílová funkce  $F$  je vždy silně nelineární funkcí ne z důvodu užití kvadrátů, ale samotné hodnoty odchylek jsou nelineárními funkcemi ladicích parametrů. Hodnotu odchylky pro konkrétní hodnoty ladicích parametrů získáme vyřešením funkce polohy mechanismu, která je vyjádřena nelineární soustavou rovnic. Vzhledem k nelinearitě funkce nelze využít metody lineární optimalizace.

Dále konstatujme, že hodnoty odchylek nejsou dány explicitně ale implicitně, protože hodnotu odchylky získáme vyřešením nelineární soustavy rovnic a proto nelze analyticky stanovit hodnotu derivace cílové funkce podle některého z ladicích parametrů. Pokud hodnotu derivace potřebujeme, musíme ji získat numericky, což je z hlediska výpočetního času drahé. Toto znamená značné omezení při výběru optimalizační metody.

Souhrnně lze říci, že cílová funkce při syntéze mechanismů je silně nelineární a nejsme schopni analyticky určit její derivace. Z tohoto také vyplývá, že terén cílové funkce bude dosti komplikovaný a proto bude mít mnoho lokálních minim, viz kap. 7.10.3.

Omezovací podmínky jsou téměř vždy nerovnostního typu, což vyplývá z jejich praktické podstaty, protože většina omezení pochází z toho, že mechanismus se může pohybovat pouze v omezeném prostoru (tzv. hledisko zástavby). Omezovací podmínky mohou být nelineární.

## 4.2 Citlivostní analýza – montážní a výrobní tolerance

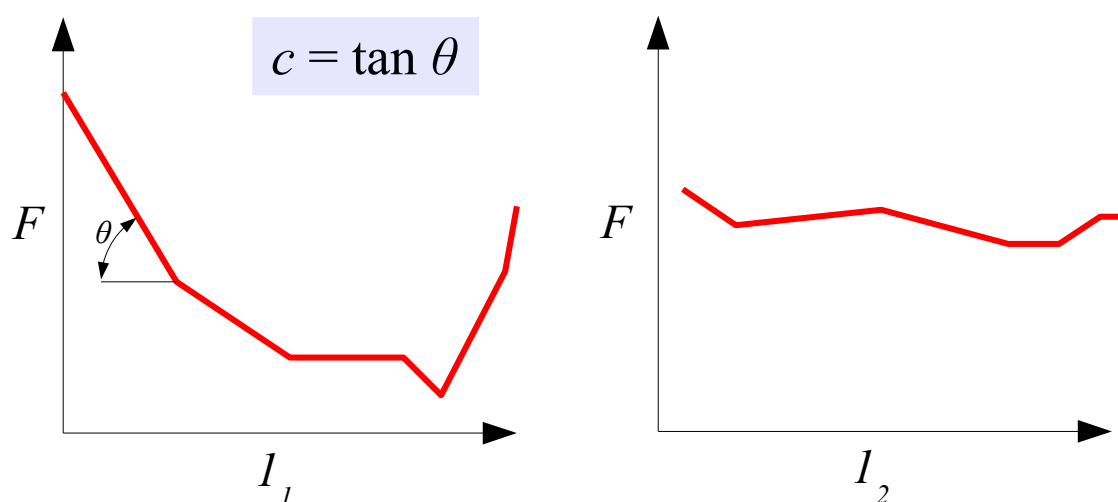
Citlivostní analýza zkoumá vliv změny některého z parametrů mechanismu na hodnotu cílové funkce  $F$ . Citlivost  $c_{li}$  cílové funkce  $F$  na změnu parametru  $l_i$  lze vyjádřit analyticky jako

$$c_{li} = \frac{\partial F}{\partial l_i}, \quad (4.8)$$

avšak většinou je tato vlastnost stanovována numericky jako

$$c_{li} = \frac{\Delta F}{\Delta l_i} = \frac{F_{(li=a)} - F_{(li=b)}}{a - b}, \quad (4.9)$$

kde  $a$  a  $b$  jsou dvě blízké hodnoty vyjadřující hodnoty parametru  $l_i$ , u kterého je stanovována citlivost. Již z definice je zřejmé, že čím větší je citlivost  $c$ , tím více lze modifikací konkrétního parametru změnit cílovou funkci, takže je zřejmé, že ladicím parametrům mechanismu by měly odpovídat vysoké citlivosti. Nejlepší interpretací citlivostní analýzy je graf, kde na vodorovné ose jsou vynášeny hodnoty zkoumaného parametru mechanismu a na svislou osu jsou vynášeny hodnoty cílové funkce. Ukázka grafické interpretace je uvedena na obr. 4.2, kde vpravo je zobrazena citlivostní analýza parametru  $l_2$ , který má slabý vliv na hodnotu  $F$ . Naopak vlevo je uveden graf pro parametr  $l_1$  jehož citlivost je vysoká, což znamená, že tento parametr je vhodným kandidátem na ladicí parametr. Hodnotu citlivosti  $c$  lze přímo v grafu sledovat pomocí sklonu křivky - čím větší sklon křivky tím vyšší citlivost.



Obrázek 4.2: Citlivostní analýza - závislost cílové funkce na parametrech mechanismu; parametr s vysokou (vlevo) a nízkou (vpravo) citlivostí

Je nutno vyzvednout, že citlivostní analýzu lze využít v technické praxi při stanovování výrobních a montážních tolerancí. Vrátime-li se k obr. 4.2, lze z pravého grafu odečíst, že pro

parametr  $l_2$  není potřeba předepisovat vysoké výrobní a montážní tolerance. Zcela naopak je tomu u parametru  $l_1$ . Dále z citlivostních grafů lze přímo určit rozsah těchto tolerancí.

Detailní citlivostní analýza mechanismu z technické praxe je provedena v kap. 8.8. Zde lze nalézt další doplňující informace.

#### 4.2.1 Výběr ladicích parametrů a jejich omezení

Úvodní postup při syntéze mechanismů lze chronologicky shrnout do následujících bodů:

1. Definice žádané vlastnosti mechanismu a tomu odpovídající cílové funkce.
2. Výběr parametrů mechanismu, které lze modifikovat, a současně rozvaha nad omezujícími podmínkami těchto parametrů mechanismu. Toto se nejčastěji děje ve spolupráci se zadavatelem nebo konstruktérem, který má představu o všech technologických a konstručních podmínkách (například hledisko zástavby).
3. Provedení citlivostní analýzy vybraných parametrů a výběr parametrů s vysokou citlivostí.
4. Opět konzultace se zadavatelem, které parametry s vysokou citlivostí jsou vhodné jako ladicí parametry (například změna některých parametrů mechanismů je značně levnější – změna svařovaných dílů je levnější než změna litých dílů).
5. Lze přistoupit k optimalizaci nalezených ladicích parametrů mechanismu.

### 4.3 Optimalizace bez omezovacích podmínek

Jak již bylo uvedeno, při optimalizaci bez omezení je cílem nalézt vektor ladicích parametrů  $\mathbf{x}_{min}$ , který minimalizuje cílovou funkci  $F$  bez toho, aniž by byla kladena jakákoli omezení na volbu ladicích parametrů, viz kap. 4.1. Matematicky lze formulaci úlohy optimalizace bez omezení definovat jako

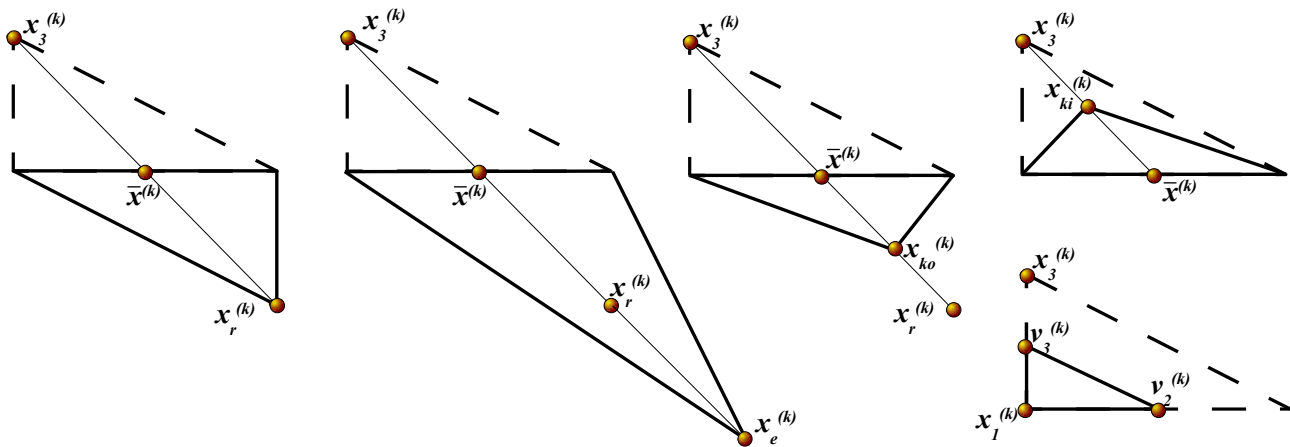
$$\mathbf{x}_{min} : \min_{\mathbf{x} \in \mathbb{R}^n} F(\mathbf{x}). \quad (4.10)$$

Metod optimalizace bez omezení je velmi široká škála. Lze využít výhradně matematické knihy jako [15], [17], [18], [19] anebo prakticky zaměřené publikace [20], [9], [10]. Po provedení numerických testů a rozboru principů a algoritmů optimalizačních metod byly vybrány dvě metody, které vykazovaly vysokou efektivitu a svými charakteristickými vlastnostmi byly vhodné pro cílové funkce, jejichž charakter je podrobně popsán v kap. 4.1. Byly vybrány dvě metody, které se principiálně liší a lze říci, že obě reprezentují dvě velké skupiny optimalizačních

metod. Metoda Nelder-Mead patří do skupiny metod přímého vyhledávání, což jsou metody, které nevyužívají derivace cílové funkce. Druhá BFGS metoda spadá do skupiny kvazinevtonovských metod, které využívají derivace cílové funkce a také historii výpočtu. Nelze jednoznačně prohlásit, která metoda je vhodnější pro danou problematiku, protože obě metody mají své nezanedbatelné přednosti, viz dále.

#### 4.3.1 Metoda simplexů neboli metoda Nelder-Mead

Základním stavebním prvkem metody je konstrukce simplexu z  $n+1$  vrcholových bodů v  $n$ -rozměrném euklidovském prostoru. Poloha jednoho vrcholu je popsána vektorem  $n$  hodnot ladicích parametrů. Simplex je nejjednodušší geometrický útvar daného  $n$ -rozměrného prostoru, takže například pro rovinu, tzn. dvourozměrný prostor, je simplexem trojúhelník, který lze jednoznačně určit pomocí tří vrcholových bodů.



Obrázek 4.3: Modifikace simplexu při dvourozměrné optimalizaci; zleva a shora: zrcadlení, zvětšení, vnější a vnitřní kontrakce, zmenšení; původní simplex je vyznačen přerušovanou čarou

Průběh hledání minima cílové funkce  $F$  je následující: nejdříve zvolíme počáteční simplex a vypočteme hodnoty cílové funkce v jeho vrcholech a uspořádáme indexy označující vrcholy simplexu podle velikosti cílové funkce. Hodnoty cílových funkcí porovnáváme a podle toho simplex zrcadlíme, zvětšujeme, zmenšujeme nebo kontrahujeme. Každé z těchto operací přísluší jistý parametr, jenž určuje rozsah dané operace, například měřítko zvětšení. Každý parametr (zrcadlení  $\rho$ , zvětšení  $\chi$ , zmenšení  $\sigma$  a kontrakce  $\gamma$ ) by měl ležet v následujících intervalech

$$\rho > 0, \chi > 1, \chi > \rho, 0 < \sigma < 1, 0 < \gamma < 1. \quad (4.11)$$

Při řešení úloh v této práci bude voleno standardní nastavení parametrů podle autorů metody simplexů profesorů Nelder a Meada



$$\rho = 1, \quad \chi = 2, \quad \sigma = \frac{1}{2}, \quad \gamma = \frac{1}{2}. \quad (4.12)$$

Ve dvourozměrném prostoru se simplexem stane trojúhelník, všechny operace uvedené v algoritmu jsou graficky vyobrazeny na následujícím obr. 4.3.

#### Algoritmus I – Metoda simplexů pro optimalizaci $n$ ladicích parametrů

Při sestavování algoritmu metody Nelder-Mead bylo čerpáno z [21], [22] a [19].

1. Položíme  $k = 0$ , zvolíme počáteční simplex, tzn.  $n + 1$  vektorů ladicích parametrů  $\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_{n+1}^{(k)}$ , kde dolní index bude označovat pořadové číslo vrcholu a horní index je pořadové číslo iterace. Dále zvolíme hodnotu  $\varepsilon_s$ , která slouží k ukončení algoritmu.
2. Vypočteme hodnoty cílových funkcí označených  $F_i^{(k)} = F(\mathbf{x}_i^{(k)})$  (pouze v případě, že jsme je dříve již nevypočetli) v jednotlivých vrcholech a provedeme přeuspořádání vektorů ladicích parametrů  $\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_{n+1}^{(k)}$  tak, aby platilo

$$F_1^{(k)} \leq F_2^{(k)} \leq \dots \leq F_{n+1}^{(k)}. \quad (4.13)$$

3. Protože hledáme minimum, můžeme říci, že bod  $\mathbf{x}_1^{(k)}$  je bodem nejvhodnějším a naopak  $\mathbf{x}_{n+1}^{(k)}$  bodem nejméně vhodným. Nyní vypočteme střední hodnotu  $n$  nejlepších bodů dle vztahu

$$\bar{\mathbf{x}}^{(k)} = \sum_{i=1}^n \frac{\mathbf{x}_i^{(k)}}{n}. \quad (4.14)$$

4. Zrcadlení; vypočteme bod zrcadlení  $\mathbf{x}_r^{(k)}$  podle níže uvedeného vztahu a vyhodnotíme cílovou funkci  $F_r^{(k)}$ , příslušející tomuto bodu. Pokud hodnota cílové funkce v bodě zrcadlení vyhoví podmínce  $F_1^{(k)} \leq F_r^{(k)} < F_{n+1}^{(k)}$ , tak přijmeme  $\mathbf{x}_r^{(k)}$  a přejdeme ke kroku 8. Pokud hodnota cílové funkce v bodě zrcadlení splňuje podmínku  $F_r^{(k)} < F_1^{(k)}$ , pokračujeme následujícím krokem 5; jinak pokračujeme krokem 6.

$$\mathbf{x}_r^{(k)} = (1 + \rho) \cdot \bar{\mathbf{x}}^{(k)} - \rho \cdot \mathbf{x}_{n+1}^{(k)} \quad (4.15)$$

5. Zvětšení; vypočteme bod zvětšení  $\mathbf{x}_e^{(k)}$  podle níže uvedeného vztahu a vyhodnotíme cílovou funkci  $F_e^{(k)}$  příslušející tomuto bodu. Pokud hodnota cílové funkce v bodě zvětšení vyhoví podmínce  $F_e^{(k)} < F_r^{(k)}$ , přijmeme  $\mathbf{x}_e^{(k)}$  a přejdeme ke kroku 8, jinak přijmeme bod  $\mathbf{x}_r^{(k)}$  a potom přejdeme ke kroku 8.

$$\mathbf{x}_e^{(k)} = (1 + \rho \cdot \chi) \cdot \bar{\mathbf{x}}^{(k)} - \rho \cdot \chi \cdot \mathbf{x}_{n+1}^{(k)} \quad (4.16)$$

6. Kontrakce; pokud platí  $F_r^{(k)} < F_n^{(k)}$  přejdeme na následující krok 7, jinak následuje:

6.1 Pokud platí podmínka  $F_n^{(k)} \leq F_r^{(k)} < F_{n+1}^{(k)}$ , pokračujeme tímto krokem, jinak přejdeme na

další krok 6.2. Vypočteme bod vnější kontrakce  $\mathbf{x}_{ko}^{(k)}$  podle níže uvedeného vztahu a vyhodnotíme cílovou funkci  $F_{ko}^{(k)}$ , příslušející tomuto bodu. Pokud  $F_{ko}^{(k)} \leq F_r^{(k)}$ , přijmeme bod  $\mathbf{x}_{ko}^{(k)}$  a přejdeme ke kroku 8, jinak přejdeme ke kroku 7.

$$\mathbf{x}_{ko}^{(k)} = (I + \rho \cdot \gamma) \cdot \bar{\mathbf{x}}^{(k)} - \rho \cdot \gamma \cdot \mathbf{x}_{n+1}^{(k)} \quad (4.17)$$

6.2 Pokud platí podmínka  $F_r^{(k)} \geq F_{n+1}^{(k)}$  pokračujeme tímto krokem, jinak přejdeme na další krok 7. Vypočteme bod vnitřní kontrakce  $\mathbf{x}_{ki}^{(k)}$  podle níže uvedeného vztahu a vyhodnotíme cílovou funkci  $F_{ki}^{(k)}$ , příslušející tomuto bodu. Pokud  $F_{ki}^{(k)} < F_{n+1}^{(k)}$ , přijmeme bod  $\mathbf{x}_{ki}^{(k)}$  a přejdeme ke kroku 8, jinak přejdeme ke kroku 7.

$$\mathbf{x}_{ki}^{(k)} = (I - \gamma) \cdot \bar{\mathbf{x}}^{(k)} + \gamma \cdot \mathbf{x}_{n+1}^{(k)} \quad (4.18)$$

7. Smrštění; vypočteme  $n$  nových vrcholů  $\mathbf{v}_2^{(k)}, \dots, \mathbf{v}_{n+1}^{(k)}$  dle níže uvedeného vztahu a vyhodnotíme cílové funkce v těchto bodech. Neuspořádané vrcholy nového simplexu  $k + 1$  budou  $\mathbf{x}_1^{(k)}, \mathbf{v}_2^{(k)}, \dots, \mathbf{v}_{n+1}^{(k)}$ . Přejdeme k dalšímu kroku 8.

$$\mathbf{v}_i = \mathbf{x}_1 + \sigma \cdot (\mathbf{x}_i - \mathbf{x}_1), \quad i = 2, \dots, n+1 \quad (4.19)$$

8. Položíme nejméně vhodný vektor ladicích parametrů  $\mathbf{x}_{n+1}^{(k)}$  roven vektoru, který jsme přijali z provedené operace (toto se neprovádí, pokud byla provedena operace zvaná smrštění). Dále vypočteme střední hodnotu ze všech bodů  $\mathbf{t}^{(k)}$ . Pokud je  $k = 0$ , položíme  $k = 1$  a pokračujeme krokem 2. Pokud je  $k > 1$ , zjistíme, zda platí  $\varepsilon_{NS} > \max |\mathbf{t}^{(k)} - \mathbf{t}^{(k-1)}|$ . V případě, že platí tato relace, algoritmus ukončíme a výsledným vektorem hodnot ladicích parametrů  $\mathbf{x}_{min}$  je vrchol s nejmenší hodnotou cílové funkce  $\mathbf{x}_1^{(k)}$  (je třeba provést uspořádání). Jinak položíme  $k = k + 1$  a postoupíme ke kroku 2.

$$\mathbf{t}^{(k)} = \sum_{i=1}^{n+1} \frac{\mathbf{x}_i^{(k)}}{n+1} \quad (4.20)$$

Pozn.: Ve výše uvedeném algoritmu není ošetřen případ, kdy by hodnoty dvou cílových funkcí si byly rovny  $F_i^{(k)} = F_{i+1}^{(k)}$ ; nastal by tak problém jak uspořádat vektory  $\mathbf{x}_i^{(k)}$ , což lze řešit implementací do algoritmu dvou pravidel pro uspořádání, které lze nalézt v [21]. Jelikož tento případ v praxi téměř nemůže nastat (pravděpodobnost, že hodnota cílové funkce bude ve dvou vrcholech stejná, je velmi malá), nebyla tato pravidla použita. Oprávněnost tohoto zjednodušení potvrzuje i skutečnost, že ani při testování tohoto algoritmu, ani při praktických výpočtech samozřejmě tato situace nenastala.

Algoritmus metody simplexů napsaný v programovacím jazyku používaném v softwaru MATLAB je přiložen v 11.3.

### 4.3.2 Kvazinewtonovské metody

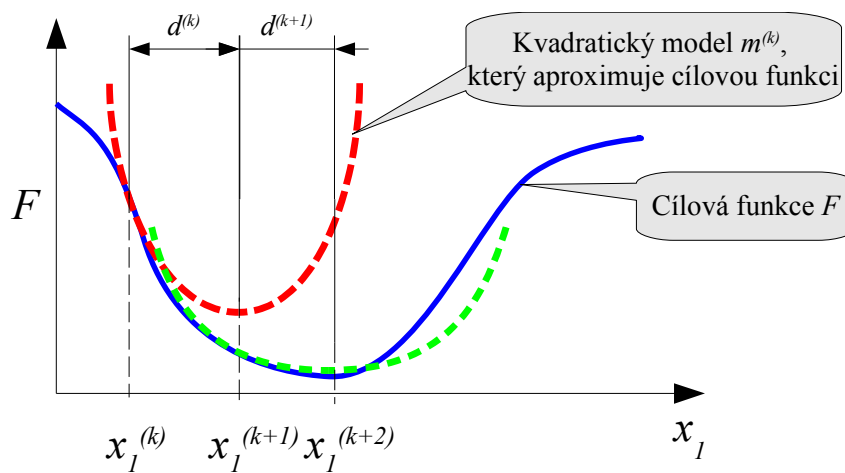
Základním stavebním prvkem této skupiny metod je konstrukce kvadratického modelu  $m^{(k)}: \mathbb{R}^n \rightarrow \mathbb{R}$  cílové funkce v  $k$ -té iteraci pomocí rozložení cílové funkce  $F$  do Taylorovy řady v bodu  $\mathbf{x}^{(k)}$  pomocí následujícího vztahu

$$F(\mathbf{x}^{(k)} + \mathbf{d}) \approx m^{(k)}(\mathbf{d}) = F^{(k)} + \nabla F^{(k)T} \cdot \mathbf{d} + \frac{1}{2} \mathbf{d}^T \cdot \mathbf{B}^{(k)} \cdot \mathbf{d}, \quad (4.21)$$

kde je použito označení  $F^{(k)} = F(\mathbf{x}^{(k)})$ , dále  $\nabla F^{(k)} \in \mathbb{R}^n$  je gradient cílové funkce v bodě  $\mathbf{x}^{(k)}$ ,  $\mathbf{d} \in \mathbb{R}^n$  je vektor (krok), který je hledán tak, aby cílová funkce  $F$  co nejvíce poklesla.  $\mathbf{B}^{(k)} \in \mathbb{R}^{n \times n}$  je matice zvaná Hessián, která má následující strukturu ( $x_i^{(k)}$  značí  $i$ -tý prvek vektoru  $\mathbf{x}^{(k)}$ )

$$\mathbf{B}_k = \begin{bmatrix} \frac{\partial^2 F^{(k)}}{\partial x_1^{(k)2}} & \frac{\partial^2 F^{(k)}}{\partial x_1^{(k)} \partial x_2^{(k)}} & \dots & \frac{\partial^2 F^{(k)}}{\partial x_1^{(k)} \partial x_n^{(k)}} \\ \frac{\partial^2 F^{(k)}}{\partial x_2^{(k)} \partial x_1^{(k)}} & \ddots & & \frac{\partial^2 F^{(k)}}{\partial x_2^{(k)} \partial x_n^{(k)}} \\ \vdots & & \ddots & \vdots \\ \frac{\partial^2 F^{(k)}}{\partial x_n^{(k)} \partial x_1^{(k)}} & \frac{\partial^2 F^{(k)}}{\partial x_n^{(k)} \partial x_2^{(k)}} & \dots & \frac{\partial^2 F^{(k)}}{\partial x_n^{(k)2}} \end{bmatrix}. \quad (4.22)$$

Pro lepší představu o principu kvazinewtonovských metod si popíšeme jednu iteraci hledání minima pro jednorozměrnou úlohu (jeden ladicí parametr  $x_1$ ) viz obr. 4.4. V tomto případě bude kvadratický model  $m^{(k)}$  reprezentován parabolou druhého stupně, což je zřejmé ze vztahu 4.21, který pro jednu dimenzi bude shodný s rovnicí kvadratické paraboly. Matice Hessián  $\mathbf{B}^{(k)}$  se pro tento případ změní na skalární hodnotu.



Obrázek 4.4: Jedna iteraci hledání minima kvazinewtonovských metod pro jednorozměrnou úlohu

9 Prvky matice  $\mathbf{B}^{(k)}$  neboli jednotlivé parciální derivace jsou vyčíslovány v bodě  $\mathbf{x}^{(k)}$ .

Postup kvazinevtonoských metod je následující: v bodě  $x_l^{(k)}$  sestrojíme na základě znalosti historie výpočtu parabolu tak, aby co nejlépe kopírovala terén cílové funkce  $F$ . Minimum této paraboly je další iterace ladicího parametru  $x_l^{(k)}$ . Tento postup opakujeme, dokud se nepřiblížíme minimu (splnění ukončovacího kritéria).

V případě, že bychom Hessián určovali exaktně, jednalo by se o Newtonovu metodu, avšak exaktně určit tuto matici je vzhledem k závěrům, které byly vyvozené z charakteru cílové funkce v kap. 4.1, nemožné, a proto bude matice aproximována, neboli jinak řečeno konstruována s využitím historie, tzn. pomocí dříve vypočtených hodnot cílové funkce  $F$ . Metody, kdy je Hessián aproximován, se nazývají kvazinevtonovské a dále se dělí podle toho, jakým způsobem je matice aproximována. Protože ve výpočtu je používána inverzní matice Hessiánu, zavedme

$$\mathbf{H}^{(k)} = (\mathbf{B}^{(k)})^{-1}. \quad (4.23)$$

Jak již bylo uvedeno inverzní matici Hessiánu získáváme pomocí historie výpočtu, dle vztahu, který navrhli autoři Broyden-Fletcher-Goldfarb-Shanno (proto metoda BFGS)

$$\mathbf{H}_{BFGS}^{(k+1)} = \mathbf{H}_{BFGS}^{(k)} + \rho^{(k)} \cdot \left( \left( \mathbf{I} + (\mathbf{y}^{(k)})^T \cdot \mathbf{H}_{BFGS}^{(k)} \cdot \mathbf{y}^{(k)} \cdot \rho^{(k)} \right) \cdot \mathbf{d}^{(k)} \cdot (\mathbf{d}^{(k)})^T - \mathbf{d}^{(k)} \cdot (\mathbf{y}^{(k)})^T \cdot \mathbf{H}_{BFGS}^{(k)} - \mathbf{H}_{BFGS}^{(k)} \cdot \mathbf{y}^{(k)} \cdot (\mathbf{d}^{(k)})^T \right),$$

$$\text{kde } \rho^{(k)} = \frac{1}{(\mathbf{d}^{(k)})^T \cdot \mathbf{y}^{(k)}}. \quad (4.24)$$

Všimněme si, že nová matice  $\mathbf{H}_{BFGS}^{(k+1)}$  je dána maticí z minulé iterace  $\mathbf{H}_{BFGS}^{(k)}$  a dalšími vstupy, které jsou vypočteny z současné iterace.

Algoritmus II - Algoritmus kvazinevtonovských metod pro optimalizaci n ladicích parametrů

Při sestavování algoritmu BFGS metody bylo čerpáno z [23] a [18].

1. Položíme  $k = 0$ , zvolíme počáteční vektor ladicích parametrů  $\mathbf{x}^{(0)}$ , kde horní index v závorce bude označovat číslo iterace. Vytvoříme počáteční inverzní Hessián  $\mathbf{H}^{(0)}$ , což musí být pozitivně definitní matice. Často se volí násobek jednotkové matice (neboli matice identity)  $\mathbf{H}^{(0)} = p\mathbf{E}$ , kde násobek  $p$  je volen  $p = 1$ ; v případě, že by optimalizační úloha nekonvergovala, lze jej zvolit menší. Naopak v případě, že se zdá první krok metody jeví jako malý, lze  $p$  volit větší. Dále zvolíme kladnou hodnotu  $\varepsilon_{BFGS} \in \mathbb{R}$ , která slouží k ukončení algoritmu a kladnou hodnotu  $h \in \mathbb{R}$ , která určuje, na jaké vzdálenosti určujeme hodnotu numerické derivace, viz *Volba kroku*  $h$  v kap. 3.3.10. Zavedme označení pro bázeový vektor  $\mathbf{e}^i \in \mathbb{R}^n$ , jehož prvky jsou

$$e_j^i = 1 \text{ pro } i=j, \quad e_j^i = 0 \text{ pro } i \neq j. \quad (4.25)$$

2. Vypočteme hodnotu gradientu cílové funkce  $\nabla F^{(0)}$  v bodě  $\mathbf{x}^{(0)}$ . Protože nemáme k dispozici analytické hodnoty prvních derivací, musíme přistoupit k numerickému výpočtu derivace,  $i$ -tý prvek gradientu cílové funkce se vypočte

$$g_i^{(0)} = \frac{F(\mathbf{x}^{(0)} + \mathbf{e}^i \cdot h) - F^{(0)}}{h}, \quad i = 1, \dots, n. \quad (4.26)$$

3. Výpočet kroku  $\mathbf{d}^{(k)} \in \mathbb{R}^n$

$$\mathbf{d}^{(k)} = -\mathbf{H}^{(k)} \cdot \mathbf{g}^{(k)}. \quad (4.27)$$

4. Nový vektor ladicích parametrů  $\mathbf{x}^{(k+1)}$  potom bude

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{d}^{(k)}. \quad (4.28)$$

Zde je nutno poznamenat, že je často doporučováno využít tzv. jednorozměrné hledání, které upřesní délku kroku, viz Poznámky k algoritmu uvedené níže níže.

5. Vypočteme gradient cílové funkce v novém bodu  $\mathbf{x}^{(k+1)}$

$$g_i^{(k+1)} = \frac{F(\mathbf{x}^{(k+1)} + \mathbf{e}^i \cdot h) - F^{(k+1)}}{h}, \quad i = 1, \dots, n. \quad (4.29)$$

6. Vypočteme rozdíl gradientů v bodech  $\mathbf{x}^{(k+1)}$  a  $\mathbf{x}^{(k)}$  a tento rozdíl označíme  $\mathbf{y}^{(k)}$

$$\mathbf{y}^{(k)} = \mathbf{g}^{(k+1)} - \mathbf{g}^{(k)}. \quad (4.30)$$

7. Nyní provedeme opravu neboli aproximaci inverzní matice Hessiánu, kterou je možné provést několika způsoby. Testovány byly tři opravy (viz níže Poznámky k algoritmu), z nichž zde uvedeme BFGS vztah, který se choval nejstabilněji

Broyden-Fletcher-Goldfarb-Shanno (dále BFGS):

$$\mathbf{H}_{BFGS}^{(k+1)} = \mathbf{H}_{BFGS}^{(k)} + \rho^{(k)} \cdot \left( \left( I + (\mathbf{y}^{(k)})^T \cdot \mathbf{H}_{BFGS}^{(k)} \cdot \mathbf{y}^{(k)} \cdot \rho^{(k)} \right) \cdot \mathbf{d}^{(k)} \cdot (\mathbf{d}^{(k)})^T - \mathbf{d}^{(k)} \cdot (\mathbf{y}^{(k)})^T \cdot \mathbf{H}_{BFGS}^{(k)} - \mathbf{H}_{BFGS}^{(k)} \cdot \mathbf{y}^{(k)} \cdot (\mathbf{d}^{(k)})^T \right), \quad (4.31)$$

$$\text{kde } \rho^{(k)} = \frac{1}{(\mathbf{d}^{(k)})^T \cdot \mathbf{y}^{(k)}}.$$

8. Pokud je  $k = 0$ , položíme  $k = 1$  a pokračujeme krokem 3. Pokud je  $k > 1$ , zjistíme, zdali platí  $\varepsilon_{BFGS} > \max \|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|$ ; v případě, že platí tato relace, algoritmus ukončíme a výsledným vektorem hodnot ladicích parametrů  $\mathbf{x}_{min} = \mathbf{x}^{(k)}$ . Jinak položíme  $k = k + 1$  a postoupíme ke kroku 3.

Algoritmus metody BFGS napsaný v programovacím jazyku používaném v softwaru MATLAB je přiložen v 11.4.

## Poznámky k algoritmu

Je nutno uvést několik poznámek k uvedenému algoritmu. Při výpočtu nového vektoru ladicích parametrů je často doporučováno využít vyhledávání podél směru definovaného krokem  $\mathbf{d}^{(k)}$ , takže pro krok 4 by se vztah změnil na

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \cdot \mathbf{d}^{(k)}, \quad (4.32)$$

kde  $\alpha^{(k)}$  je skalární součinitel, který zpřesňuje délku kroku  $\mathbf{d}^{(k)}$ . Ke hledání součinitele  $\alpha^{(k)}$  jsou používány tzv. linesearch algoritmy, kterých lze nalézt mnoho a které jsou založeny na mnoha různých principech. Autorem byly testovány<sup>10</sup> tyto algoritmy:

- Wolfe linesearch algoritmus (algoritmus 3.2 z [15]) využívající zoom (algoritmus 3.3 z [15])
- Quadratic fit search (algoritmus 13c [19])
- Wolfe-Powell algoritmus (vývojový diagram 2.5.3 z [17])
- Backtracking-Armijo linesearch [24]

Nejvíce se osvědčil poslední uvedený algoritmus. Jeho použitím se sice snížil počet iterací algoritmu BFGS, ale počet vyčíslení cílové funkce, který udává efektivitu algoritmu, se zvýšil, a proto doporučujeme linesearch techniku vynechat.

Dále je nutno doplnit, že byly testovány tři vztahy pro opravu matice inverzního Hessiánu, a to Davidon-Fletcher-Powell z [18], Broyden-Fletcher-Goldfarb-Shanno z [23] a modifikovaná Broyden-Fletcher-Goldfarb-Shanno z [23]. Oprava dle Davidon-Fletcher-Powell se ukázala nejméně efektivní a v jistých případech nekonvergovala. Vztah pro modifikovanou Broyden-Fletcher-Goldfarb-Shanno opravu byl velice efektivní, ale nestabilní (v několika málo případech nekonvergoval). Proto bylo rozhodnuto, že bude doporučena sice méně efektivní oprava BFGS, ale zato stabilní. Vztahy pro výše uvedené opravy matic jsou uvedeny v příloze 11.2.

Zajímavým poznatkem je, že je vhodné kombinovat BFGS s modifikovanou BFGS opravou, a to tak, že pro počátečních několik iterací (cca 9) byla použita formule BFGS a dále byla použita již modifikovaná BFGS formule, viz příloha 11.2.1.

Provedené testy jsou uvedeny v příloze 11.2.2.

<sup>10</sup> Z důvodu rozsahu nejsou testy publikovány v práci.

### 4.3.3 Srovnání metod

Aplikace obou algoritmů na praktický příklad je uvedena v kap. 7.9.

#### **Shrnutí předností a nedostatků metody simplexů**

Nelder-Mead je jedna z nejpobulárnějších metod patřících do skupiny přímého prohledávání, což znamená, že během prohledávání terénu cílové funkce nevyužíváme první ani vyšší derivace cílové funkce. Toto se stává velkou předností právě v případě, kdy je určení derivace respektive derivací obtížné či dokonce nemožné. Právě v případě řešení zdvihových závislostí složitých mechanismů lze určit derivace pouze numericky, což je časově náročné (pro výpočet numerického gradientu je zapotřebí  $n + 1$  vyčíslení cílové funkce), a proto bylo k této metodě přistoupeno. Další neopomenutelnou předností, která je cenná zvláště u inženýrských aplikací, je robustnost této metody, pro kterou není překážkou ani složitý terén cílové funkce, na kterém mohou některé metody založené na sofistikovanějších matematických principech ztroskotat.

Naopak jednou z podstatných nevýhod metody je, že nijak nevyužívá historii, tzn. nijak nevyužívá hodnot cílové funkce, které byly dříve vypočteny. Jinými slovy lze říci, že metoda nekonstruuje žádný model cílové funkce na rozdíl od další zmíněné skupiny kvazinevtonovských metod. Bohužel je tento nedostatek metody simplexů v případě optimalizace tvaru trajektorie bodu vodicího mechanismu také podstatný, protože každý výpočet cílové funkce je časově náročný a proto není efektivní tuto hodnotu použít pouze jednou.

Protože si tyto dvě základní vlastnosti trochu protiřečí, byla využita ještě jedna skupina metod, a to kvazinevtonovské metody. Obě metody jsou na závěr této kapitoly numericky porovnány.

#### **Shrnutí předností a nedostatků kvazinevtonovských metod**

Přednosti a nedostatky jsou opačné na rozdíl od metody simplexů. Výhodou je, že využíváme historii, tzn. že pomocí vypočtených hodnot cílových funkcí stavíme kvadratický model funkce. Metoda BFGS se v testech ukázala logicky efektivnější než metoda simplexů. Naopak její nevýhodou je, že není tolik robustní jako předchozí metoda.

## 4.4 Optimalizace s omezovacími podmínkami

Formulaci úlohy optimalizace s omezovacími podmínkami již byla popsána v kap. 4.1. Připomeňme formulaci úlohy optimalizace s omezením – optimalizace je minimalizace cílové funkce prostřednictvím ladicích parametrů tak, abychom současně dodrželi podmínky omezení. Označíme-li vektor  $n$  ladicích parametrů jako  $\mathbf{x} \in \mathbb{R}^n$  a cílovou funkci jako  $F: \mathbb{R}^n \rightarrow \mathbb{R}$ , lze formulovat úlohu matematické optimalizace s omezením jako hledání optimálního vektoru ladicích parametrů, který označíme  $\mathbf{x}_{min}$ , což lze zapsat jako

$$\mathbf{x}_{min} : \min_{\mathbf{x} \in \mathbb{R}^n} F(\mathbf{x}) \quad \text{omezeno} \quad \begin{matrix} h_i(\mathbf{x}) \leq 0, \text{ kde } i=1 \dots r \\ g_j(\mathbf{x}) = 0, \text{ kde } j=1 \dots s \end{matrix}, \quad (4.33)$$

kde  $h_i(\mathbf{x})$  jsou nerovnostní podmínky omezení a  $g_j(\mathbf{x})$  jsou rovnostní podmínky omezení a  $r$  a  $s$  označují jejich počty.

Při výběru metody optimalizace je nutno připomenout charakter cílové funkce, která je při syntéze mechanismů silně nelineární a nejsme schopni analyticky určit její derivace, a charakter omezení, která jsou většinou nerovnostního typu  $h_i(\mathbf{x})$  a mohou být nelineární, viz Charakteristika cílových funkcí a omezovacích podmínek v kap. 4.1.

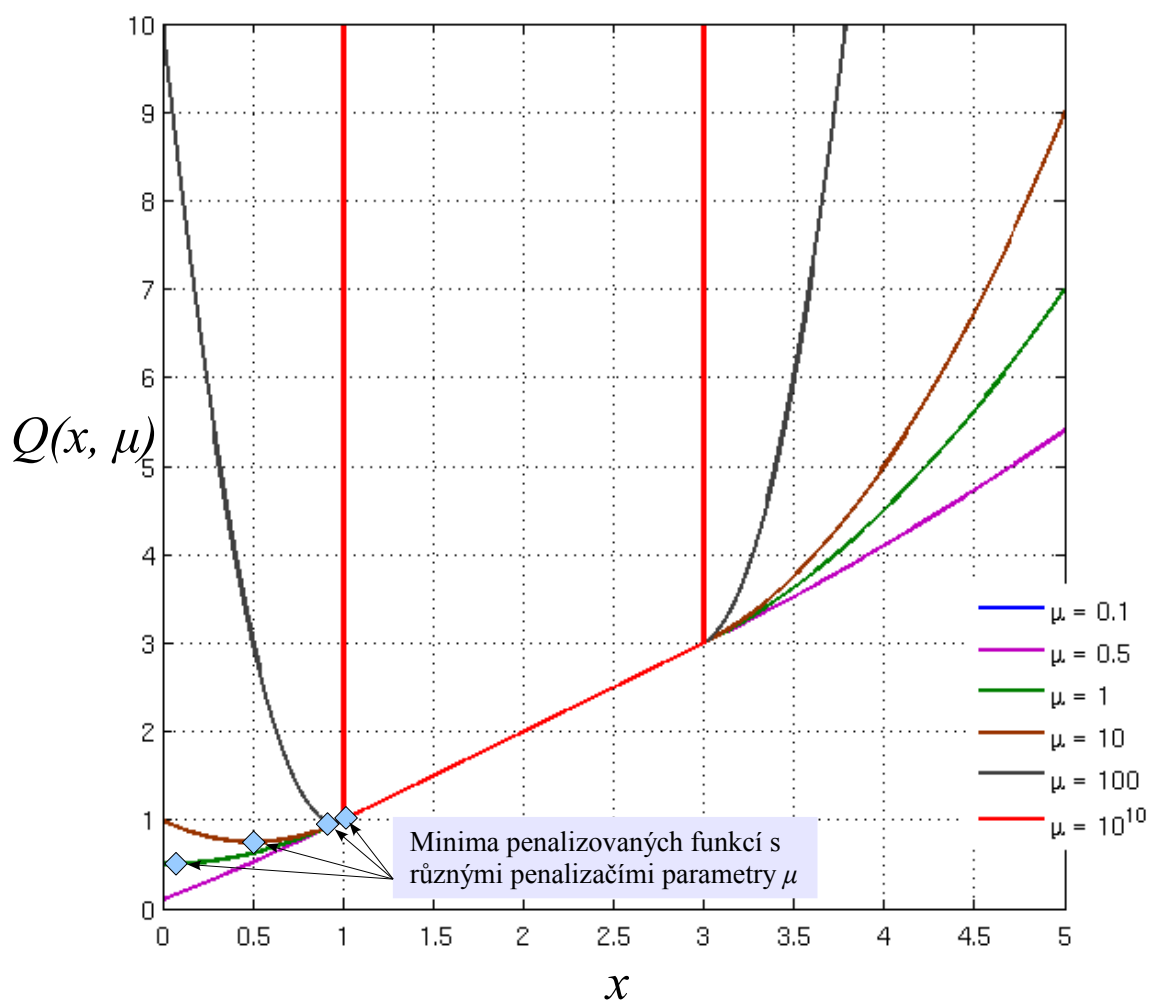
S ohledem na uvedenou charakteristiku v úvahu přidají tyto metody:

1. Metoda penalizačních funkcí – metoda je jednoduchá a stabilní, která umožňuje řešit všechny typy úloh. Nevýhodou je, že nalezený vektor ladicích parametrů může mírně porušovat omezovací podmínky. Metoda je detailně popsána v následující kap. 4.4.1.
2. Metoda bariérových funkcí – metoda neumožňuje řešit problémy s rovnostními omezeními a již z jejího principu je její řešení špatně podmíněno a vyskytují se zde často konvergenční potíže.
3. Metoda rozšířených Lagrangiánů – implementace nerovností dosti komplikuje algoritmus a navíc jsou vnášeny další neznámé do úlohy. Efektivita algoritmu by měla být vyšší než u obou předchozích.
4. Genetické algoritmy – principiálně se liší od předchozích metod. Hledají globální minima na předem definované oblasti ladicích parametrů. Daň za hledání globálního minima je vysoká časová náročnost. Jejich implementace je značně komplikovaná. Jejich užití je prezentováno v rámci multibody



Při výběru algoritmu bylo voleno mezi metodou penalizace a metodou rozšířených Lagrangiánů. Vzhledem k tomu, že při testech na úlohách z technické praxe penalizační metoda konvergovala během několika mála iterací (v jistých případech dokonce stačila jedna iterace) nebylo třeba hledat metodu s vyšší efektivitou a navíc algoritmus metody rozšířených Lagrangiánů je mnohem komplikovanější. Nedostatek metody penalizace spočívající v tom, že nalezený vektor ladicích parametrů může mírně porušovat omezovací podmínky, lze eliminovat přísnější volbou omezovacích podmínek nebo změnou nastavení parametrů algoritmu.

#### 4.4.1 Metoda penalizačních funkcí



Obrázek 4.5: Průběhy penalizovaných funkcí pro různé penalizační parametry

Metoda penalizace převádí úlohu optimalizace s omezením definovanou v 4.33 na úlohu bez omezení. Toto je provedeno rozšířením cílové funkce  $F$  o penalizační funkce, jejichž

hodnota je tím větší, čím více je porušena omezovací podmínka. Aby nevznikaly nehladké funkce, je výhodné použít kvadratické penalizační funkce. Cílová funkce rozšířená o penalizační funkci má tvar

$$Q(\mathbf{x}, \mu) = F(\mathbf{x}) + \mu \cdot \sum_{j=1}^s g_j^2(\mathbf{x}) + \mu \cdot \sum_{i=1}^r [\max(0, h_i(\mathbf{x}))]^2, \quad (4.34)$$

kde skalár  $\mu$  je penalizační parametr a člen  $\max(0, h_i(\mathbf{x}))$  značí výběr větší ze dvou hodnot tak, aby byla funkce penalizována pouze pokud dojde k porušení nerovnostní podmínky.

Optimalizační úlohu s omezením 4.33 nyní po penalizaci zformulujeme jako optimalizační úlohu bez omezení s cílovou funkcí  $Q(\mathbf{x}, \mu)$  následovně

$$\mathbf{x}_{min} : \min_{\mathbf{x} \in \mathbb{R}^n, \mu \in \mathbb{R}} Q(\mathbf{x}, \mu). \quad (4.35)$$

Pro nalezení optimálního vektoru ladicích parametrů  $\mathbf{x}_{min}$  lze nyní využít jakoukoli metodu optimalizace bez omezení. Doporučujeme využít metodu BFGS prezentovanou v kap. 4.3.2. Využití této metody se při numerických pokusech ukázalo nejvhodnější.

Abychom metodou penalizace získali uspokojivé výsledky je nezbytné podrobně porozumět jejímu charakteru a chování. Popíšme některá úskalí této metody na triviálním příkladě definovaném následovně

$$x_{min} : \min_{x \in \mathbb{R}} F(x) = x \quad \text{omezeno} \quad \begin{matrix} x-3 \leq 0 \\ -x+1 \leq 0 \end{matrix}. \quad (4.36)$$

Při vykreslení funkcí (viz obr. 4.5) je zřejmé, že řešením této úlohy je

$$x_{min} = 1. \quad (4.37)$$

Po zavedení penalizace je úloha převedena na minimalizaci funkce

$$Q(x, \mu) = x + \mu \cdot [\max(0, x-3)]^2 + \mu \cdot [\max(0, -x+1)]^2. \quad (4.38)$$

Nyní diskutujme nad volbou hodnoty penalizačního parametru  $\mu$ . Na obr. 4.5 je vykreslen průběh funkce  $Q(x, \mu)$  pro různé hodnoty tohoto parametru. Všimněme si, že polohy minim funkce  $Q(x, \mu)$  leží v nepřipustné oblasti, ve které porušují jednu z nerovnostních podmínek

$$x-3 \leq 0. \quad (4.39)$$

Lze vysledovat, že čím více zvyšujeme parametr  $\mu$ , tím více se blíží poloha minima k přípustné oblasti. Pro

$$\mu \rightarrow \infty \quad (4.40)$$

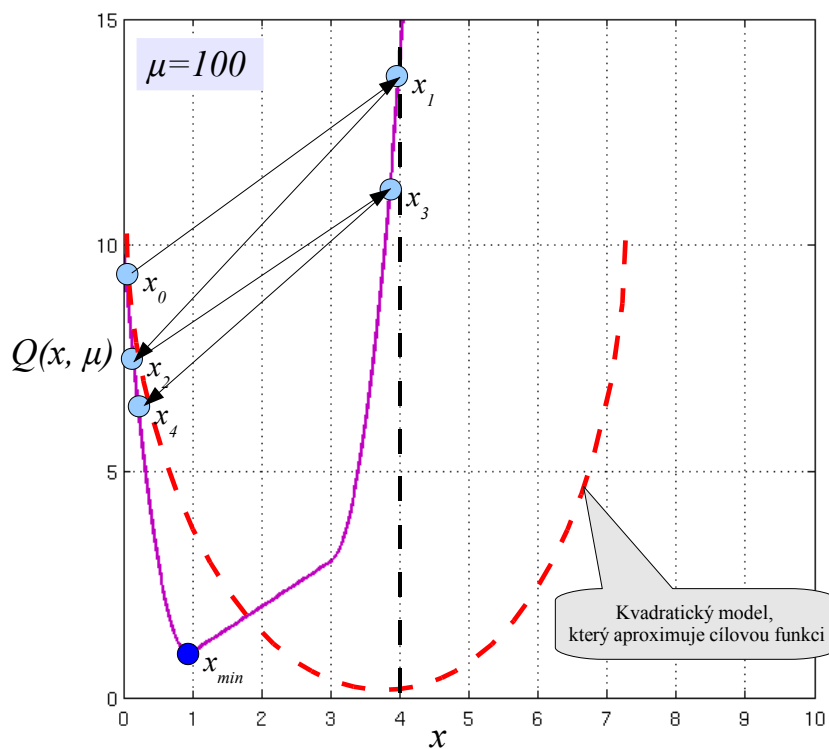
by minimum funkce  $Q(x, \mu)$  odpovídalo analytickému řešení 4.37. Z těchto vlastností penalizované

funkce by se mohlo zdát, že je výhodné dosadit za  $\mu$  co nejvyšší hodnotu, což však není zcela správné.

Zkoumejme blíže průběh optimalizačního procesu funkce  $Q(x, \mu)$  s vysokou hodnotou penalizačního parametru - například

$$\mu = 100. \quad (4.41)$$

Průběh takovéto funkce  $Q(x, \mu)$  je vykreslen fialovou křivkou na obr. 4.6. Všimněme si, že v nepřipustných oblastech má funkce velice strmý charakter, což způsobuje při optimalizaci značné potíže. Uvažujme, že k hledání minima funkce použijeme Newtonovu nebo některou kvazinetonovskou metodu. Tato metoda hledá minimum funkce tak, že nahrazuje cílovou funkci kvadratickým modelem a řešením jedné iterace je minimum tohoto kvadratického modelu, viz obr. 4.4 a přílehlý popis.



Obrázek 4.6: Průběh hledání minima při vysoké hodnotě penalizačního parametru

V našem příkladu se jedná o minimalizaci jedné proměnné, takže kvadratický model bude mít podobu paraboly. Proces optimalizace by byl pro tento případ následující. Počáteční odhad řešení  $x_0$  se nalézá za přípustnou oblastí, kde je funkce  $Q(x, \mu)$  značně strmá. Aproximujeme-li v tomto bodě funkci tečně parabolou, tak získáme parabolu značných rozměrů, jejíž minimum  $x_1$  je dalece vzdáleno skutečnému řešení  $x_{min}$ . Bod  $x_1$  se opět vyskytuje v nepřipustné a současně strmé oblasti a proto by parabola, jež by nahrazovala cílovou funkci, byla opět rozsáhlá a její minimum by

bylo jen o málo blíže ke skutečnému minimu  $\mathbf{x}_{min}$  než počáteční odhad  $\mathbf{x}_0$ . Takto bychom mohli pokračovat dále a je zcela zřejmé, že takovýto proces není efektivní. Z tohoto vyplývá, že volba příliš vysokého penalizačního parametru není vhodná.

Nyní je zřejmé, že k volbě penalizačního parametru je třeba přistupovat s jistou opatrností. Logický přístup je takový, že nejprve je zvoleno malé  $\mu$  a pro toto je provedena optimalizace. Po optimalizaci je otestováno, zda je řešení s jistou tolerancí v přípustné oblasti. Pokud ano, tak jsme získali řešení, a pokud ne, tak zvýšíme penalizační parametr  $\mu$  a opakujeme optimalizaci. Tímto jsme popsali celý algoritmus, který je prezentován níže.

#### Algoritmus III – Metoda penalizační funkce

1. Rozšíříme cílovou funkci o penalizační funkce dle

$$Q(\mathbf{x}, \mu) = F(\mathbf{x}) + \mu \cdot \sum_{j=1}^s g_j^2(\mathbf{x}) + \mu \cdot \sum_{i=1}^r [\max(0, h_i(\mathbf{x}))]^2, \quad (4.42)$$

2. Nastavíme pořadové číslo iterace  $k = 0$ , maximální počet iterací  $maxit$ , nastavíme toleranci porušení omezovacích podmínek  $v$ , faktor  $\beta > 1$ , který je použit pro násobné zvyšování penalizačního parametru. Zvolíme malý penalizační parametr  $\mu^{(k)} > 0$ , např. 0,1. Odhadneme počáteční vektor ladicích parametrů  $\mathbf{x}^{(k)}$ .
3. Nalezneme vektor ladicích parametrů  $\mathbf{x}^{(k+1)}$ , který minimalizuje funkci  $Q(\mathbf{x}, \mu^{(k)})$  s počátečním odhadem  $\mathbf{x}^{(k)}$ . Toto provedeme některou z metod optimalizace bez omezení, např. algoritmus II nebo I.
4. Nyní ověříme, zda se vektor  $\mathbf{x}^{(k)}$  nalézá v přípustné oblasti s tolerancí  $v$

$$\begin{aligned} \text{pokud všechna } \left[ \begin{array}{l} h_i(\mathbf{x}^{(k)}) \leq v \\ |g_j(\mathbf{x}^{(k)})| \leq v \end{array} \right] &\Rightarrow \mathbf{x}^{(k)} \text{ je řešení} \rightarrow konv = 0 \text{ ukonči algoritmus,} \\ \text{pokud } k = maxit &\Rightarrow \text{úloha nekonverguje} \rightarrow konv = 1 \text{ ukonči algoritmus,} \\ \text{jinak} &\rightarrow \mu^{(k+1)} = \beta \cdot \mu^{(k)} \quad k = k + 1 \rightarrow \text{opakuj postup od bodu 2.} \end{aligned} \quad (4.43)$$

Pozn.: Vzhledem k tomu, že výsledný vektor ladicích parametrů může porušovat omezení v rámci předepsané tolerance  $v$ , je vhodné toto porušení po výpočtu zkontrolovat. Dosazením výsledného vektoru  $\mathbf{x}_{min}$  do podmínek omezení, zjistíme, zda se nalézáme těsně na hranici některé z podmínek omezení apod.

## **Shrnutí a doporučení pro metodu penalizačních funkcí**

Metoda penalizačních funkcí je svou podstatou velice jednoduchá a tím umožňuje uživateli její plné pochopení, v čemž je její největší síla. Nespornou výhodou metody je její obecná použitelnost, která umožňuje řešení jakýchkoliv úloh optimalizace s omezením. Dalším kladem je její jednoduše implementovatelný a stabilní algoritmus.

Naopak mezi nevýhody metody patří skutečnost, že výsledný vektor ladicích parametrů může ležet v nepřipustné oblasti. Tuto skutečnost lze však jednoduše eliminovat tím, že zpřísníme omezující podmínky a zvolíme toleranci porušení omezovací podmínky tak, že výsledné ladicí parametry budou zaručeně ležet v pro nás přípustné oblasti.

Dalším nedostatkem metody je její efektivita, která je nižší než například u metody rozšířených Lagrangiánů, avšak v našich praktických aplikacích syntézy mechanismů jsme se s tímto nepotýkali. Výsledek se dostavil vždy během několika mála iterací, v jistých případech i během první iterace.

Praktická aplikace metody a její srovnání s multibody softwarem je uvedeno v kap. 7.10.1.

## **4.5 Lokální a globální minimum cílové funkce**

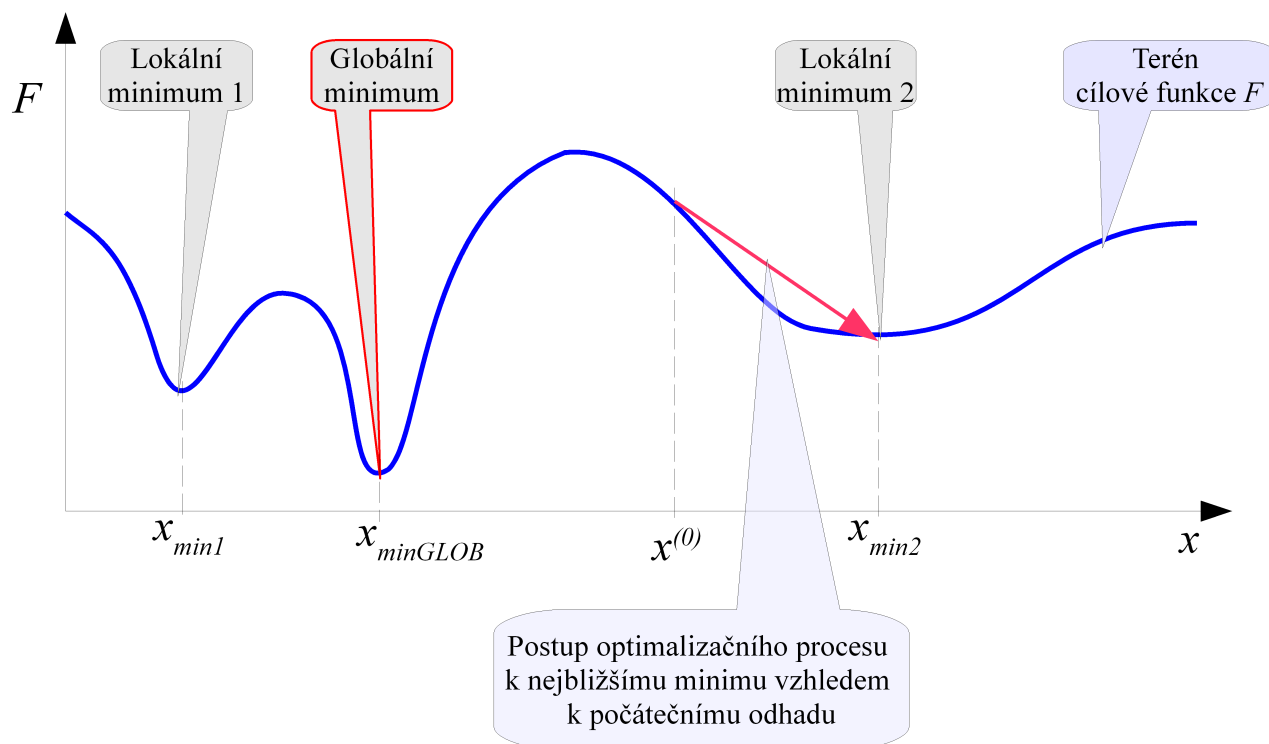
Nejprve připomeňme, že na konci kap. 4.1 bylo v charakteristice cílové funkce stanoveno, že terén cílové funkce v problematice syntézy mechanismů je značně složitý a proto má mnoho lokálních minim.

Zmiňme se o lokálních a globálních extrémech cílové funkce. Je nezbytné si uvědomit, že výsledkem optimalizačního procesu nemusí být nutně globální neboli celkové minimum. Spíše naopak; téměř vždy je výsledkem některé lokální minimum cílové funkce. Z obr. 4.7 je zřejmé, že algoritmus nalezne často to minimum, které je k počátečnímu odhadu vektoru ladicích parametrů nejbližší.

Žádná optimalizační metoda nemůže zaručit nalezení globálního minima. Některé algoritmy jsou založeny na navzorkování určité oblasti, čímž vznikne síť bodů a v těchto určují hodnoty cílové funkce. K těmto metodám patří například genetické algoritmy nebo screening, viz kap. 7.9.5. Ovšem ani tento postup nezaručí nalezení globálního minima, protože toto minimum může ležet mimo předem nadefinovanou oblast a také nelze provést navzorkování oblasti nekonečně husté. Dlužno dodat, že časová náročnost těchto metod je značně vyšší než u metod prezentovaných

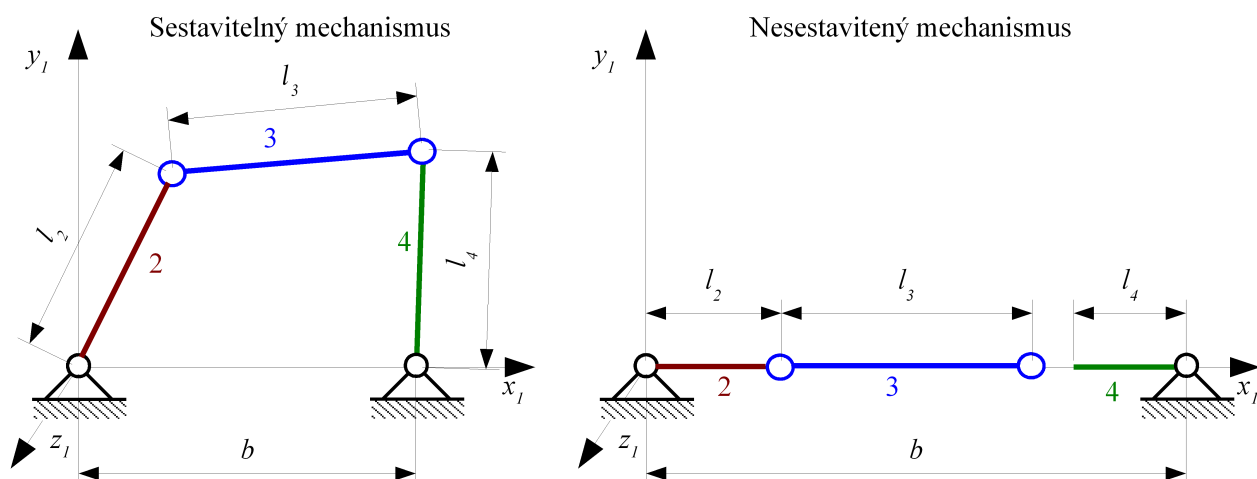
zde. Podrobné informace o hledání globálního minima lze nalézt v [45].

Pro řešení tohoto problému lze navrhnout jednoduché doporučení, které spočívá v tom, že lze provést optimalizaci několikrát pro různé počáteční odhady vektoru ladících parametrů. Toto zajistí prohledání větší oblasti.



Obrázek 4.7: Lokální a globální minima cílové funkce a jejich vliv na optimalizaci

## 4.6 Nesestavitelnost mechanismů během optimalizačního procesu



Obrázek 4.8: Čtyřkloubový mechanismus sestavitelná a nesestavitelná varianta

Pojmem nesestavitelnost mechanismů rozumíme mechanismus, jehož rozměry nedovolují tento fyzicky sestavit. Popišme to na triviálním příkladu čtyřkloubového rovinného

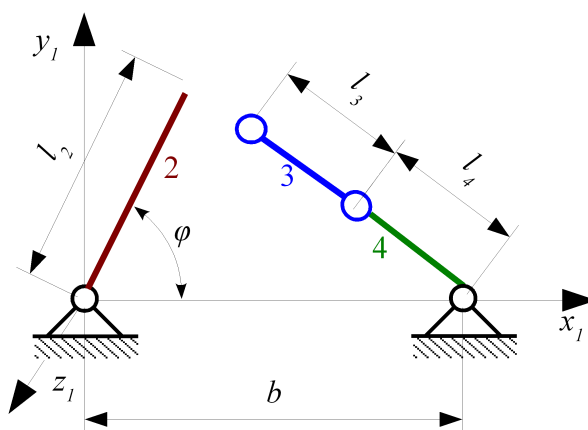
mechanismu, který je uveden na obr. 4.8. Je zřejmé, že rozměry tohoto mechanismu musí odpovídat podmínce sestavitelnosti, která má tvar

$$l_2 + l_3 + l_4 \geq b. \quad (4.44)$$

Pokud tato podmínka není splněna nelze mechanismus pro tyto konkrétní hodnoty rozměrů mechanismus sestavit.

Další variantou nesestavitelnosti mechanismu je varianta zobrazená na obr. 4.9. Jedná se o případ, kdy sestavení mechanismu je znemožněno hodnotou souřadnice mechanismu, což je v tomto případě úhel  $\varphi$ . Podmínka sestavitelnosti by v tomto případě vyplývala z cosinové věty

$$l_3 + l_4 \geq \sqrt{l_2^2 + b^2 - 2 \cdot b \cdot l_2 \cdot \cos \varphi}. \quad (4.45)$$



Obrázek 4.9: Čtyřkloubový mechanismus nesestavitelný z důvodu hodnoty souřadnice mechanismu

Pokud bychom se pokusili vyřešit úlohu polohy nesestavitelného mechanismu, neměla by tato úloha neměla řešení. V praxi bychom to zjistili tak, že při výpočtu úlohy polohy některým z řešičů nelineárních soustav rovnic (viz kap. 5) by tento buď překročil maximální dovolený počet iterací, nebo porušil podmínku 5.55. Konkrétně v kap. 5.8.4 je podrobněji popsáno chování řešičů nelineárních soustav rovnic při analýze nesestavitelných mechanismů a také je odvozena dodatečná podmínka 5.55, kterou lze nesestavitelnost mechanismu identifikovat během několika počátečních iterací. Připomeňme, že v algoritmu VII, který je autor doporučuje pro řešení úlohy polohy, je stanovována výchozí proměnná  $konv$ , která nabývá těchto hodnot

$$\begin{cases} konv=0 & \text{úlohy polohy konvergovala k řešení} \Rightarrow \text{sestavitelný mechanismus,} \\ \left\{ \begin{array}{l} konv=1 \text{ překročen maximální počet iterací } itmax \\ konv=2 \text{ nebylo vyhověno podmínce 5.55} \end{array} \right\} & \Rightarrow \text{nesestavitelný mechanismus.} \end{cases} \quad (4.46)$$

Nyní vysvětleme, proč se tématikou nesestavitelnosti zde zabýváme. Při optimalizačním procesu v oblasti syntézy mechanismů mohou nastávat případy, kdy optimalizační algoritmus navrhne ladící proměnné tak, že právě vznikne nesestavitelný mechanismus, což má za následek, že

úloha polohy nekonverguje, protože hodnota cílové funkce nemůže být vyčíslena a optimalizační algoritmus následně havaruje. Existuje několik možností, jak toto řešit, viz [12], [25] nebo [26].

### Volba doplňkových omezujících podmínek

Zcela přirozenou a primární možností se jeví volba doplňkových omezujících nerovnostních nebo rovnostních podmínek, které řešitel navrhne tak, aby byl mechanismus sestavitelný. Tento způsob je velice jednoduše implementovatelný s využitím některé optimalizační metody s omezením.

Je vhodnější pro optimalizaci s omezení využít takovou optimalizační metodu, která navrhuje ladicí parametry výhradně z přípustné oblasti, viz str. 466 v [17], str. 500 v [15] nebo str. 833 v [19]. Do této oblasti metod spadá například metoda bariérových funkcí. Metoda penalizačních funkcí volí ladicí parametry i z nepřípustné oblasti, viz 4.4.1, a proto do těchto metod nespadá a je méně vhodná pro toto použití. Při přísnější volbě omezujících podmínek a vyšší hodnotě penalizačního parametru  $\mu$  je použitelná.

Je zřejmé, že omezovacími podmínkami nelze obsáhnout všechny podmínky sestavitelnosti, protože není reálné, aby řešitel všechny tyto podmínky našel. Lze doplnit pouze jednotlivé podmínky sestavitelnosti, které vznikly na základě zkušeností z předchozího havarovaného optimalizačního procesu.

Pokud bychom chtěli zahrnout všechny podmínky sestavitelnosti, tak bychom museli zvolit jako omezující podmínky celou úlohu polohy. Toto je však značně složité výpočetně velice drahé a neefektivní. U optimalizačního problému definovaného v 4.33 doplníme omezujícími podmínkami

$$\mathbf{x}_{min} : \min_{\mathbf{x} \in \mathbb{R}^n} F(\mathbf{x}) \quad \text{omezeno} \quad \left\{ \begin{array}{l} h_i(\mathbf{x}) \leq 0, \text{ kde } i=1 \dots r \\ g_j(\mathbf{x}) = 0, \text{ kde } j=1 \dots s \\ |\mathbf{f}(\mathbf{q}_{(x)})| - \varepsilon_{nes} \leq 0 \end{array} \right\}, \quad (4.47)$$

o člen

$$|\mathbf{f}(\mathbf{q}_{(x)})| - \varepsilon_{nes} \leq 0, \quad (4.48)$$

který vyjadřuje podmínku splnění úlohy polohy, viz 3.2.5, kde  $\mathbf{q}_{(x)}$  vyjadřuje vektor neznámých souřadnic vazeb a  $(\mathbf{x})$  připomíná, že tento vektor je funkčně závislý na ladicích parametrech. Skalár  $\varepsilon_{nes}$  je tolerance splnění úlohy polohy, viz kap. 5. Nutno upozornit, že tato tolerance by měla být volena vyšší než tolerance  $\varepsilon_{NS}$  ukončující algoritmus výpočtu úlohy polohy, viz kap. 5. Optimalizační problém 4.47 by bylo opět vhodné řešit některou z metod, která navrhuje ladicí



parametr pouze z dovolené oblasti, například metodou bariérových funkcí (viz str. 466 v [17], str. 500 v [15] nebo str. 833 v [19]).

Podrobné informace k problematice sestavitelnosti mechanismů během optimalizačního procesu a konkrétně metodu, která vychází z podobných principů jako předdefinovaný optimalizační problém 4.47, lze detailně rozpracovanou lze nalézt v [25].

## 5 Metody řešení soustav nelineárních rovnic

V kapitole 3.2.5 bylo popsáno, že poloha mechanismu je určena soustavou nelineárních, transcendentních a skalárních rovnic, viz vztah 3.4. V kap. 3.3.8 bylo vysvětleno, jak tuto soustavu rovnic sestavit. Zde se budeme zabývat řešením této soustavy. Soustavu 3.4 přepíšeme do následující formy

$$\begin{array}{ccc} f_1(\mathbf{s}, \mathbf{l}, \mathbf{q})=0, & f_1(\mathbf{x})=0, & \\ f_2(\mathbf{s}, \mathbf{l}, \mathbf{q})=0, & f_2(\mathbf{x})=0, & \\ \vdots & \vdots & \\ f_{r-6}(\mathbf{s}, \mathbf{l}, \mathbf{q})=0, & f_n(\mathbf{x})=0, & \end{array} \quad \Rightarrow \quad \mathbf{f}(\mathbf{x})=\mathbf{0}, \quad (5.1)$$

kde  $\mathbf{s}$  označuje vektor vstupních souřadnic,  $\mathbf{l}$  vektor rozměrů mechanismu,  $\mathbf{q}$  vektor neznámých souřadnic vazeb a  $n$  je počet rovnic. Z matematického hlediska jsou  $\mathbf{s}$  a  $\mathbf{l}$  konstanty a proto je v zápisu budeme dále vynechávat. Z důvodu matematických zvyklostí a obecnosti jsme neznámé souřadnice vazeb  $\mathbf{q}$  v této kapitole označili jako vektor  $\mathbf{x} \in \mathbb{R}^n$ . Celá soustava bude souhrnně značena jako

$$\mathbf{f}(\mathbf{x})=\mathbf{0}. \quad (5.2)$$

Jedná se o soustavu  $n$  nelineárních rovnic. Funkce  $\mathbf{f}$  je vektorovou funkcí, což lze matematicky zapsat jako

$$\mathbf{f}(\mathbf{x}): \mathbb{R}^n \rightarrow \mathbb{R}^n. \quad (5.3)$$

V praxi lze velice zřídka nalézt řešení této soustavy analytickými metodami (například dosazovací metodou viz [27]), a proto je nutné řešit tuto soustavu numericky. Dále budou navrženy a srovnány metody a jejich algoritmy pro řešení těchto soustav.

Metody řešení jsou založeny na tom, že na soustavu rovnic 5.2 nahlížíme jako na vektorovou funkci dle 5.3. Výsledkem této funkce je vektor reziduálů  $\mathbf{r}$

$$\mathbf{f}(\mathbf{x})=\mathbf{r}, \quad (5.4)$$

pokud by se vektor  $\mathbf{r}$  rovnal nulovému vektoru, pak by vektor  $\mathbf{x}$  byl řešením soustavy. Nejprve uvedme kostru všech algoritmů prezentovaných metod řešení soustav nelineárních rovnic.

Algoritmus IV – Algoritmus zobecňující metody řešení soustav nelineárních rovnic

Algoritmy všech metod řešení soustav nelineárních rovnic lze shrnout takto:

1. Nastavíme  $k = 0$ , maximální počet iterací  $maxit$  a zvolíme hodnotu  $\varepsilon_{NS}$ , která definuje limitní normu reziduálů sloužící pro ukončení výpočtu. Odhadneme

počáteční vektor  $\mathbf{x}^k$ , a to nejlépe tak, aby tento vektor byl blízko hledanému řešení.

2. Algoritmus navrhne podle určitého vztahu vektor neznámých  $\mathbf{x}$ . Jednotlivé metody se liší právě tímto vztahem, kterým určujeme  $\mathbf{x}$ .
3. Navržený vektor neznámých  $\mathbf{x}$  se dosadí do

$$\mathbf{f}(\mathbf{x}) = \mathbf{r}, \quad (5.5)$$

čímž se vypočte vektor reziduálů  $\mathbf{r}$ . Čím nižší hodnotu budou mít jeho prvky, tím blíže je vektor  $\mathbf{x}$  k řešení soustavy.

4. Nyní pomocí normy vypočteme skalární hodnotu, která charakterizuje celkovou velikost vektoru reziduálů  $\mathbf{r}$  neboli délku tohoto vektoru jako

$$|\mathbf{r}| = \sum_{j=1}^k r_j^2. \quad (5.6)$$

5. Hodnotu normy  $|\mathbf{r}|$  srovnáme s námi předepsanou přesností  $\varepsilon_{NS}$  a pokud je vypočtená hodnota  $|\mathbf{r}|$  nižší než  $\varepsilon_{NS}$ , pokládáme vektor  $\mathbf{x}$  za řešení soustavy 5.2. Pokud je tomu naopak, se vracíme do bodu 1 a opakujeme celý cyklus.

$$\begin{aligned} \text{pokud } |\mathbf{r}| \leq \varepsilon &\Rightarrow \mathbf{x} \text{ je řešením soustavy} \rightarrow konv=0 \text{ ukonči algoritmus,} \\ \text{pokud } k = \maxit &\Rightarrow \text{úloha nekonverguje} \rightarrow konv=1 \text{ ukonči algoritmus,} \\ \text{jinak } k = k + 1 &\text{ a jdi zpět do bodu 1.} \end{aligned} \quad (5.7)$$

Pozn.:  $konv$  je skalární, výstupní charakteristika algoritmu, která udává informaci jakým způsobem byl algoritmus ukončen a zda našel řešení. Charakteristika  $konv$  nabývá následujících hodnot

$$\begin{aligned} konv=0 &\text{ algoritmus konvergoval a našel řešení,} \\ konv=1 &\text{ algoritmus divergoval a dosáhl maximálního počtu iterací } \maxit, \\ konv=2 &\text{ algoritmus divergoval a byl ukončen z důvodu zvyšování délky kroku } p \text{ viz 5.55} \end{aligned} \quad (5.8)$$

## 5.1 Newtonova metoda

Mějme libovolný vektor  $\mathbf{x}$ , který není řešením soustavy  $n$  rovnic 5.2. Newtonova metoda spočívá v hledání vektoru přírůstků  $\mathbf{p}$  tak, aby vektor  $\mathbf{x} + \mathbf{p}$  byl řešením soustavy rovnic, takže musí platit, že

$$\mathbf{f}(\mathbf{x} + \mathbf{p}) = \mathbf{0}. \quad (5.9)$$

Vektor přírůstků  $\mathbf{p}$  se často nazývá Newtonův krok nebo pouze krok.

Funkci  $\mathbf{f}(\mathbf{x})$  aproximujeme pomocí Taylorova rozvoje prvního řádu

$$\mathbf{f}(\mathbf{x} + \mathbf{p}) = \mathbf{f}(\mathbf{x}) + \mathbf{J}(\mathbf{x}) \cdot \mathbf{p} + \mathbf{O}(|\mathbf{p}|^2), \quad (5.10)$$

kde  $\mathbf{J}(\mathbf{x})$  je Jacobiho matice, jejíž prvky jsou hodnoty parciálních derivací funkce  $\mathbf{f}(\mathbf{x})$  podle jednotlivých prvků vektoru  $\mathbf{x}$ . Matice má tvar

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}. \quad (5.11)$$

$\mathcal{O}(|\mathbf{p}|^2)$  vyjadřuje v 5.10 je chybová funkce, která nahrazuje prvky vyššího řádu Taylorova rozvoje. Vzhledem k tomu, že tyto prvky nahrazené funkcí  $\mathcal{O}(|\mathbf{p}|^2)$  zanedbáme, lze prohlásit, že chyba aproximace je úměrná kvadrátu přírůstku  $|\mathbf{p}|^2$ .

Shrňme: hledáme krok  $\mathbf{p}$  tak, aby byl výsledkem soustavy rovnic, takže musí platit 5.9 a současně funkci aproximujeme vztahem 5.10. Při zanedbání chybové funkce  $\mathcal{O}(|\mathbf{p}|^2)$  a dosazení z 5.9 do 5.10 získáme výsledný vztah Newtonovy metody pro neznámý vektor přírůstku  $\mathbf{p}$

$$\mathbf{J}(\mathbf{x}) \cdot \mathbf{p} = -\mathbf{f}(\mathbf{x}). \quad (5.12)$$

Tento vztah zapojíme do iterační smyčky popsané v algoritmu IV a tím získáme algoritmus metody. Existuje zde však podstatná komplikace, a to je výpočet Jacobiho matice. Vzhledem k typu a rozsahu našich úloh by bylo značně náročné nebo spíše prakticky nemožné stanovit tuto matici exaktně, a proto musíme matici  $\mathbf{J}$  stanovit numericky pomocí vztahů pro numerický výpočet derivace. Vztah pro prvek Jacobiho matice je následující

$$J_{ij} = \frac{\partial f_i}{\partial x_j} = \frac{f_i(\mathbf{x} + \mathbf{e}^j \cdot h) - f_i(\mathbf{x})}{h}, \text{ kde } i, j = 1, 2, \dots, n, \quad (5.13)$$

kde  $\mathbf{e}^j$  je báze vektor a horní index vektoru označuje který prvek je jednotkový. Matematický zápis je následující

$$\begin{aligned} e_j^i &= 1 \text{ pro } i=j, \\ e_j^i &= 0 \text{ pro } i \neq j, \end{aligned} \quad ; \text{ např. } \mathbf{e}^2 = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}. \quad (5.14)$$

Parametrem  $h$  je ve vztahu 5.14 označena vzdálenost, na které určíme hodnotu numerické derivace. Pravidla pro volbu hodnoty  $h$  lze nalézt na konci kap. 3.3.10.

Na základě všech výše uvedených poznatků lze sestavit algoritmus Newtonovy metody.<sup>11</sup>

<sup>11</sup> Zavedme následující značení:  $k$  je číslo iterace,  $\mathbf{x}^k$  je vektor, který je uvažován v  $k$ -té iteraci jako možné řešení soustavy,  $\mathbf{J}^k$  je Jacobiho matice sestavená pro  $\mathbf{x}^k$ ,  $\mathbf{f}^k$  je vyčíslení funkce 5.4 pro  $\mathbf{x}^k$ .

Zavedme následující značení:  $k$  je číslo iterace,  $\mathbf{x}^k$  je vektor, který je uvažován v  $k$ -té iteraci jako možné řešení soustavy,  $\mathbf{J}^k$  je Jacobiho matice sestavená pro  $\mathbf{x}^k$ ,  $\mathbf{f}^k$  je vyčíslení funkce 5.4 pro  $\mathbf{x}^k$ .

#### Algoritmus V – Newtonova metoda pro řešení soustav nelineárních rovnic

1. Nastavíme  $k = 0$ , maximální počet iterací  $maxit$ , hodnotu  $h$  dle kap. 3.3.10 a zvolíme hodnotu  $\varepsilon_{NS}$ , která definuje limitní normu reziduálů sloužící pro ukončení výpočtu. Odhadneme vektor  $\mathbf{x}^k$ , a to nejlépe tak, aby tento vektor byl blízko hledanému řešení.
2. Vyčíslíme  $\mathbf{f}^k$ .
3. Matici  $\mathbf{J}^k$  stanovíme dle vztahu 5.13, a to pomocí dvou do smyček

$$\begin{array}{l} \text{pro } i=1,2 \dots n \\ \quad \text{pro } j=1,2 \dots n \\ \quad \quad J_{ij}^k = \frac{f_i(\mathbf{x}^k + \mathbf{e}^j \cdot h) - f_i^k}{h}, \\ \quad \text{konec smyčky} \\ \text{konec smyčky} \end{array} \quad (5.15)$$

4. Stanovíme krok  $\mathbf{p}$  dle vztahu 5.12. Toto lze provést jakoukoli metodou pro řešení soustav lineárních rovnic. Při malém  $n$  postačuje výpočet pomocí inverzní matice

$$\mathbf{p} = -(\mathbf{J}^k)^{-1} \cdot \mathbf{f}^k. \quad (5.16)$$

5. Vypočteme nový vektor  $\mathbf{x}^{k+1}$

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{p}. \quad (5.17)$$

6. Vyčíslíme  $\mathbf{f}^{k+1}$ , čímž současně získáme hodnotu reziduálů dle 5.4 a následovně stanovíme normu tohoto vektoru, kterou srovnáme s hodnotou  $\varepsilon_{NS}$

$$\begin{array}{l} \text{pokud } |\mathbf{f}^{k+1}| \leq \varepsilon_{NS} \Rightarrow \mathbf{x}^k \text{ je řešení soustavy rovnic} \rightarrow konv=0 \text{ ukonči algoritmus,} \\ \text{pokud } k=maxit \Rightarrow \text{úloha nekonverguje} \rightarrow konv=1 \text{ ukonči algoritmus,} \\ \text{jinak } k=k+1 \rightarrow \text{opakuji postup od bodu 3.} \end{array} \quad (5.18)$$

### Zhodnocení Newtonovy metody

Metoda patří k tradičním a osvědčeným metodám řešení soustav nelineárních rovnic. Stejně jako u všech ostatních metod je největší část výpočtového času spotřebována na vyčíslování vektorové funkce  $\mathbf{f}$ . Proto je počet vyčíslení logickým měřítkem efektivity metody. Vzhledem k tomu, že nejsme schopni vyjádřit analyticky parciální derivace funkce  $\mathbf{f}$ , je zřejmou nevýhodou

metody nutnost numerického výpočtu Jacobiho matice, kdy na jedno sestavení této matice připadá  $n$  výpočtů celé vektorové funkce  $\mathbf{f}$ . Srovnání s ostatními metodami je uvedeno v kap. 5.9.

## 5.2 Broydenova dobrá metoda

Broydenova metoda spadá do skupiny kvazinevtonovských metod, které nevyčísľují Jacobiho matici přímo dle její definice 5.11, ale snaží se tuto matici sestavit pomocí určitého vztahu, který využívá historii výpočtu. Pro lepší představu o Broydenově metodě lze zjednodušeně poznamenat, že se jedná o známou jednorozměrnou metodu sečen zobecněnou pro řešení soustavy  $n$  nelineárních rovnic.

C. G. Broyden odvodil a navrhl na str. 581 v [28] následující výraz

$$\mathbf{J}^{k+1} = \mathbf{J}^k + \frac{(\mathbf{y}^k - \mathbf{J}^k \cdot \mathbf{p}^k) \cdot (\mathbf{p}^k)^T}{(\mathbf{p}^k)^T \cdot \mathbf{p}^k}, \text{ kde } \mathbf{y}^k = \mathbf{f}^{k+1} - \mathbf{f}^k. \quad (5.19)$$

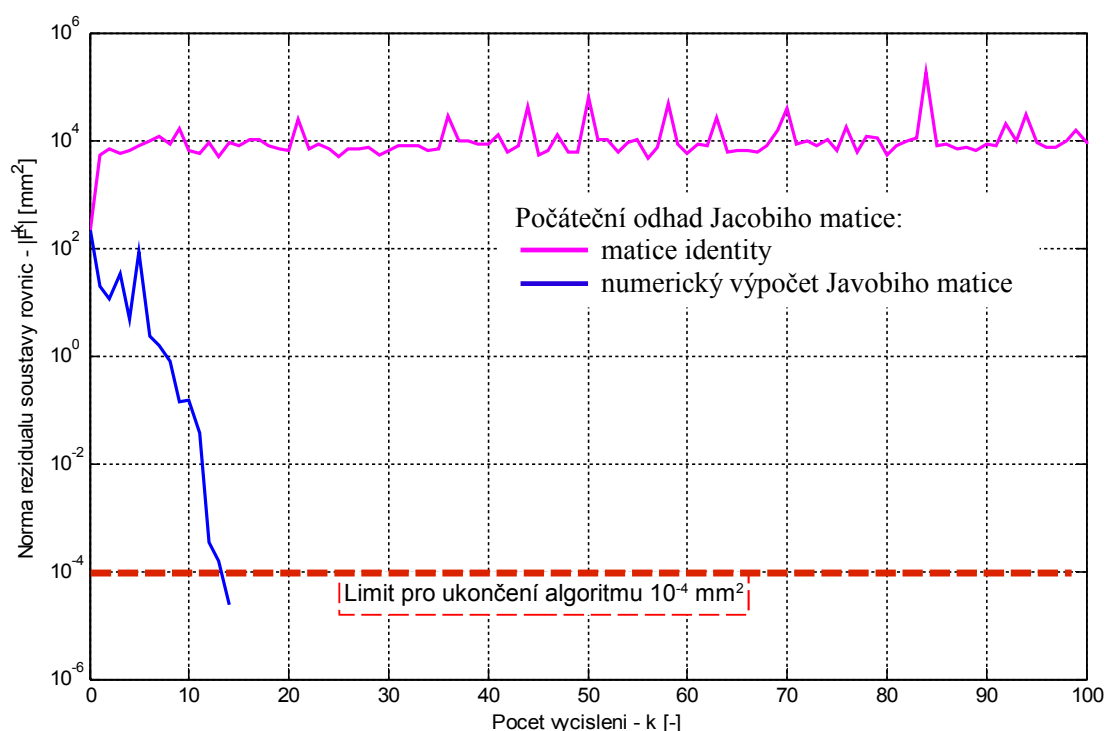
Jedná se o formuli Broydenovy dobré metody. Existuje také vztah pro Jacobiho matici, který se mírně liší od 5.19 a který se nazývá Broydenova špatná metoda<sup>12</sup>, viz kap. 5.3. Pomocí Sherman-Morrisonova vztahu [29] odvodil Broyden formuli, která stanovuje přímo inverzní matici Jacobiho matice, kterou budeme značit  $\mathbf{H}$

$$(\mathbf{J}^{k+1})^{-1} = \mathbf{H}^{k+1} = \mathbf{H}^k + \frac{(\mathbf{p}^k - \mathbf{H}^k \cdot \mathbf{y}^k) \cdot (\mathbf{p}^k)^T \cdot \mathbf{H}^k}{(\mathbf{p}^k)^T \cdot \mathbf{H}^k \cdot \mathbf{y}^k}, \quad (5.20)$$

tímto se zbavuje mírné komplikace s řešením maticové rovnice, což je popsáno v algoritmu V v bodě 4. Pokud se zaměříme na vztah pro inverzní Jacobiho matici 5.20, lze vidět, že nová matice  $\mathbf{H}^{k+1}$  je tvořena na základě znalosti matice  $\mathbf{H}^k$  z minulé iterace, kroku  $\mathbf{p}$  a rozdílu hodnot funkce  $\mathbf{f}$ . Matice  $\mathbf{H}^{k+1}$  je stavěna pomocí historie výpočtu a není zde potřeba nadbytečného a drahého vyčíslení funkce  $\mathbf{f}$ .

Jistou komplikací této metody je, že na počátku potřebujeme odhad Jacobiho matice  $\mathbf{J}^0$ . Použití pro odhad  $\mathbf{J}^0$  matici identity se v praxi ukázalo jako zcela nepřipustné. Při využití jednotkové matice jako počátečního odhadu Broydenova dobrá metoda divergovala, viz obr. 5.1. Z grafu lze vypořozovat že trend křivky, která zobrazuje hodnotu normy reziduálů, je konstantní tedy divergentní. Jako nezbytné se ukázalo na počátku výpočtu odhadnout Jacobiho matici pomocí numerického výpočtu derivace jak je uvedeno ve vztahu 5.13. Při použití takového odhadu úloha konverguje během několika málo vyčíslení funkce  $\mathbf{f}$ , viz obr. 5.1.

<sup>12</sup> Názvy Broydenova dobrá a špatná metoda vznikly 70. letech, kdy se matematici domnívali, že dobrá metoda je tou výkonnější z těchto dvou. V roce 1982 brazilský matematik v [30] zjistil, že za jistých okolností je tomu naopak, viz kap. 5.3. Názvy dobrá a špatná metoda již zůstaly nezměněny.



Obrázek 5.1: Porovnání kovergence Broydenovy dobré metody pro různé počáteční odhady Jacobiho matice

Vztah 5.20 společně s poznatky uvedenými v předchozí kapitole pro Newtonovu metodu nám umožňuje nyní sestavit algoritmus Broydenovy dobré metody.<sup>11</sup>

Algoritmus VI – Broydenova dobrá metoda pro řešení soustav  $n$  nelineárních rovnic

1. Nastavíme  $k = 0$ , maximální počet iterací  $maxit$ , hodnotu  $h$  dle kap. 3.3.10 a zvolíme hodnotu  $\varepsilon_{NS}$ , která definuje limitní normu reziduálů sloužící pro ukončení výpočtu. Odhadneme vektor  $\mathbf{x}^k$ , a to nejlépe tak, aby tento vektor byl blízko hledanému řešení.
2. Vyčíslíme  $\mathbf{f}^k$ .
3. Stanovíme počáteční odhad Jacobiho matice  $\mathbf{J}^k$  dle vztahu 5.13, a to pomocí dvou do smyček

$$\begin{aligned}
 &\text{pro } i=1,2 \dots n \\
 &\quad \text{pro } j=1,2 \dots n \\
 &\quad \quad J_{ij}^k = \frac{f_i(\mathbf{x}^k + \mathbf{e}^j \cdot h) - f_i^k}{h}, \\
 &\quad \text{konec smyčky} \\
 &\quad \text{konec smyčky}
 \end{aligned} \tag{5.21}$$

4. Výpočet inverzní matice k Jacobiho matici

$$\mathbf{H}^k = (\mathbf{J}^k)^{-1}. \quad (5.22)$$

5. Stanovíme krok  $\mathbf{p}$

$$\mathbf{p}^k = -\mathbf{H}^k \cdot \mathbf{f}^k. \quad (5.23)$$

6. Vypočteme nový vektor  $\mathbf{x}^{k+1}$

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{p}^k. \quad (5.24)$$

7. Vyčíslení  $\mathbf{f}^{k+1}$ .

8. Vyčíslení  $\mathbf{y}^k$

$$\mathbf{y}^k = \mathbf{f}^{k+1} - \mathbf{f}^k. \quad (5.25)$$

9. Výpočet nové inverzní Jacobiho matice

$$\mathbf{H}^{k+1} = \mathbf{H}^k + \frac{(\mathbf{p}^k - \mathbf{H}^k \cdot \mathbf{y}^k) \cdot (\mathbf{p}^k)^T \cdot \mathbf{H}^k}{(\mathbf{p}^k)^T \cdot \mathbf{H}^k \cdot \mathbf{y}^k} \quad (5.26)$$

10. Vyčíslíme  $\mathbf{f}^{k+1}$ , čímž současně získáme hodnotu normy reziduálů dle 5.4. Tuto hodnotu srovnáme s hodnotou  $\varepsilon_{NS}$

pokud  $|\mathbf{f}^{k+1}| \leq \varepsilon_{NS} \Rightarrow \mathbf{x}^k$  je řešení soustavy rovnic  $\rightarrow konv=0$  ukonči algoritmus,  
 pokud  $k = \maxit \Rightarrow$  úloha nekonverguje  $\rightarrow konv=1$  ukonči algoritmus,  
 jinak  $k = k + 1 \rightarrow$  opakuj postup od bodu 5. (5.27)

V dostupné literatuře lze často najít doporučení (např. [15]), že lze zvýšit efektivitu a stabilitu Broydenovy metody zavedením tzv. „linesearch“ algoritmů. Vzhledem k tomu, že Kvazinewtonovské metody nepočítají s exaktní Jacobiho maticí, ale s její aproximací, lze předpokládat, že délka Newtonova kroku  $\mathbf{p}$  navržená algoritmem není optimální délkou z hlediska hledání řešení soustavy rovnic. Tento nedostatek je nahrazován hledáním skalárního násobiče  $\alpha$  ve směru kroku  $\mathbf{p}$ , takže vektor  $\mathbf{x}^k$  lze vyjádřit jako

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha \cdot \mathbf{p}. \quad (5.28)$$

Jinými slovy lze říci, že hledáme hodnotu násobiče  $\alpha$  tak, aby se reziduály funkce  $\mathbf{f}$  blížily co nejvíce nule. Algoritmus Broydenovy dobré metody s jednorozměrným vyhledáváním by nabyl následující podoby.

Algoritmus VI – Broydenova dobrá metoda pro řešení soustav  $n$  nelineárních rovnic s implementací jednorozměrného hledání

Algoritmus je totožný s algoritmem 3; liší se pouze v bodu 6, který vypadá následně:

6. Hledáme násobič  $\alpha$  tak, aby byla minimalizována norma reziduálů neboli funkce  $\mathbf{f}$



$$\alpha_{min} : \min |f(x^k + \alpha \cdot p)|, \text{ potom } x^{k+1} = x^k + \alpha_{min} \cdot p. \quad (5.29)$$

Tuto operaci provádí „linesearch“ algoritmy, které jsou popsány v [15], [17], [19] nebo [24].

## Zhodnocení Broydenovy dobré metody

Broydenova metoda se zdá být přirozeně vhodná pro použití při řešení úloh polohy mechanismů a podobných technických problémů, které jsou matematicky vyjádřeny soustavou nelineárních rovnic a nelze u nich analyticky určit Jacobiho matici. Ve srovnání s Newtonovou metodou je zde nespornou výhodou, že Jacobiho matice je stavěna na základě znalosti historie výpočtu a tedy bez přílišného vyčíslování funkce  $f$ , které je potřeba při numerickém výpočtu parciálních derivací. Nabízí se rozhodnutí, zda využít „linesearch“ algoritmy. Na základě srovnání efektivity metod z kap. 5.9 navrhuje tyto metody nevyužívat. Připouštíme, že využitím „linesearch“ se algoritmus může stávat stabilnějším, ale tato domněnka se v našich příkladech a dostupných zkušenostech nepotvrdila. Srovnání s ostatními metodami je uvedeno v kap. 5.9.

## 5.3 Martinezova hybridní metoda

Jak již bylo zmíněno u vztahu 5.19 navrhl C. G. Broyden v [28] dvě formule pro výpočet aproximované Jacobiho matice. Z těchto byla využívána pouze jedna, protože vztah pro Broydenovu špatnou metodu byl považována za méně efektivní. J. M. Martinez v [30] zjistil, že za jistých podmínek je Broydenova špatná metoda výkonnější. Vznikla Martinezova hybridní metoda, která je kombinací Broydenovy dobré a špatné metody. Vzhledem k tomu, že metoda se liší pouze ve výběru vztahu pro aproximaci inverzní Jacobiho matice  $H$ , bude uveden přímo algoritmus bez dalšího popisu principu metody; tento lze najít v předchozí kap. 5.2.

### Algoritmus VII – Martinezova hybridní metoda

1. Nastavíme  $k = 0$ , maximální počet iterací  $maxit$ , poměrnou toleranci vůči délce prvního kroku  $tol$ , hodnotu  $h$  dle kap. 3.3.10 a zvolíme hodnotu  $\varepsilon_{NS}$ , která definuje limitní normu reziduálů sloužící pro ukončení výpočtu. Odhadneme vektor  $x^k$ , a to nejlépe tak, aby tento vektor byl blízko hledanému řešení.
2. Vyčíslíme  $f^k$ .
3. Stanovíme počáteční odhad Jacobiho matice  $J^k$  dle vztahu 5.13.
4. Výpočet inverzní matice k Jacobiho matici

$$\mathbf{H}^k = (\mathbf{J}^k)^{-1}. \quad (5.30)$$

5. Stanovíme krok  $\mathbf{p}$

$$\mathbf{p}^k = -\mathbf{H}^k \cdot \mathbf{f}^k. \quad (5.31)$$

6. Kontrola délky kroku  $\mathbf{p}$  dle pravidla odvozeného v kap. 5.8.4

$$\text{pokud } |\mathbf{p}^{k+1}| > \text{tol} \cdot |\mathbf{p}^0| \Rightarrow \text{úloha nekonverguje} \rightarrow \text{konv}=2 \rightarrow \text{ukonči algoritmus.} \quad (5.32)$$

7. Vypočteme nový vektor  $\mathbf{x}^{k+1}$

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{p}^k. \quad (5.33)$$

8. Vyčíslení  $\mathbf{f}^{k+1}$ .

9. Vyčíslení  $\mathbf{y}^k$

$$\mathbf{y}^k = \mathbf{f}^{k+1} - \mathbf{f}^k. \quad (5.34)$$

10. Výpočet nové inverzní Jacobiho matice dle Broydenovy dobré nebo špatné metody

$$\text{pokud platí } k > 0 \text{ a současně } \frac{|(\mathbf{p}^k)^T \cdot \mathbf{p}^{k-1}|}{(\mathbf{p}^k)^T \cdot \mathbf{H}^k \cdot \mathbf{y}^k} \geq \frac{|(\mathbf{y}^k)^T \cdot \mathbf{y}^{k-1}|}{(\mathbf{y}^k)^T \cdot \mathbf{y}^k}$$

$$\mathbf{H}^{k+1} = \mathbf{H}^k + \frac{(\mathbf{p}^k - \mathbf{H}^k \cdot \mathbf{y}^k) \cdot (\mathbf{y}^k)^T}{(\mathbf{y}^k)^T \cdot \mathbf{y}^k}, \quad (5.35)$$

jinak

$$\mathbf{H}^{k+1} = \mathbf{H}^k + \frac{(\mathbf{p}^k - \mathbf{H}^k \cdot \mathbf{y}^k) \cdot (\mathbf{p}^k)^T \cdot \mathbf{H}^k}{(\mathbf{p}^k)^T \cdot \mathbf{H}^k \cdot \mathbf{y}^k}.$$

11. Vyčísleme  $\mathbf{f}^{k+1}$ , čímž současně získáme hodnotu normy reziduálů dle 5.4. Tuto hodnotu srovnáme s hodnotou  $\varepsilon_{NS}$

$$\begin{aligned} \text{pokud } |\mathbf{f}^{k+1}| \leq \varepsilon_{NS} &\Rightarrow \mathbf{x}^k \text{ je řešení soustavy rovnic} \rightarrow \text{konv}=0 \text{ ukonči algoritmus,} \\ \text{pokud } k = \text{maxit} &\Rightarrow \text{úloha nekonverguje} \rightarrow \text{konv}=1 \text{ ukonči algoritmus,} \\ \text{jinak } k = k+1 &\rightarrow \text{opakuj postup od bodu 5.} \end{aligned} \quad (5.36)$$

Také v Martinezově metodě by bylo možné využít „linesearch“ algoritmy pro upřesnění délky kroku  $\mathbf{p}$ , ale na základě zkušeností a úvah z předchozí kap. 5.2, tuto implementaci nepublikujeme.

## Zhodnocení Martinezovy hybridní metody

Metoda je založena na Broydenově metodě a proto pro ni platí podobné zhodnocení jak již bylo uvedeno na konci kap. 5.2. Oproti Broydenově dobré metodě je však algoritmus

Martinezovy metody složitější, ale efektivnější.

Srovnání s ostatními metodami je uvedeno v kap. 5.9.

## 5.4 Trust-Region metody

Významnou skupinou metod sloužících k řešení soustav nelineárních rovnic jsou tzv. „Trust-Region“ metody (dále TR). I v této skupině metod aproximujeme funkci  $f$ , která označuje soustavu nelineárních rovnic, pomocí Taylorova rozvoje prvního řádu

$$f(\mathbf{x}^k + \mathbf{p}) \approx f(\mathbf{x}^k) + \mathbf{J}^k \cdot \mathbf{p}, \text{ kde } \mathbf{f}^k = f(\mathbf{x}^k), \mathbf{J}^k = \mathbf{J}(\mathbf{x}^k). \quad (5.37)$$

Stejně jako v Newtonově metodě vycházíme z toho, že máme vektor  $\mathbf{x}^k$  ke kterému hledáme krok  $\mathbf{p}$  tak, aby platilo, že

$$f(\mathbf{p}) = \mathbf{0}. \quad (5.38)$$

TR metody nahlíží na hledání řešení soustavy rovnic jako na optimalizační úlohu, kde hledáme minimum cílové funkce. Cílová funkce musí být postavena tak, aby její minimum bylo současně řešením soustavy rovnic 5.2 - tomuto odpovídá následující cílová funkce

$$m^k(\mathbf{p}) = \frac{1}{2} (\mathbf{f}(\mathbf{p})) \cdot (\mathbf{f}(\mathbf{p}))^T, \quad (5.39)$$

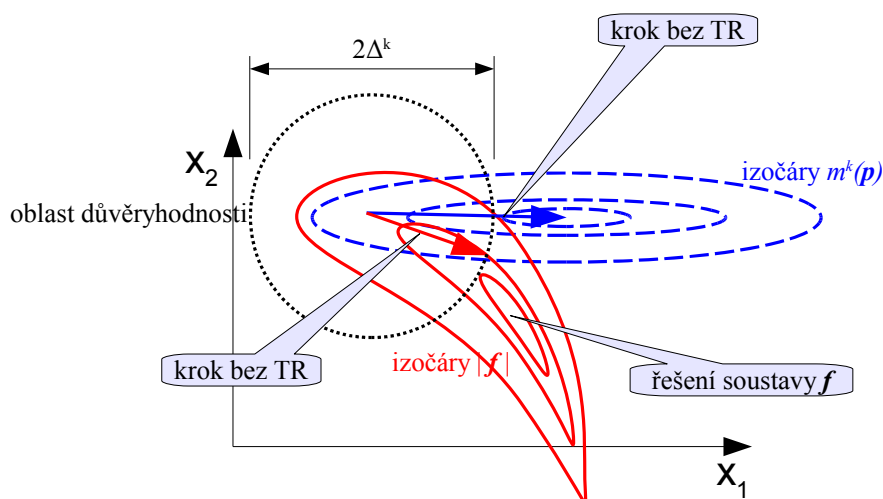
kde se lze snadnou úvahou přesvědčit, že pokud existuje řešení soustavy rovnic, tak globální minimum cílové funkce  $m^k(\mathbf{p})$  je řešením soustavy rovnic a v tomto bodě platí

$$m^k(\mathbf{p}) = 0. \quad (5.40)$$

Rovnici 5.39 lze dosazením 5.37 a umocněním rozepsat do tvaru

$$m^k(\mathbf{p}) = \frac{1}{2} \mathbf{f}^k \cdot (\mathbf{f}^k)^T + \mathbf{p}^T \cdot (\mathbf{J}^k)^T \cdot \mathbf{f}^k + \frac{1}{2} \mathbf{p}^T \cdot (\mathbf{J}^k)^T \cdot \mathbf{J}^k \cdot \mathbf{p}. \quad (5.41)$$

Klíčovou myšlenkou TR metod je, že aproximace soustavy rovnic uvedená v 5.37 je platná pouze v určité oblasti, která je definována jako poloměr  $\Delta^k$  (oblast důvěryhodnosti). Což je logická úvaha založená na tom, že v 5.37 je hodnota  $\mathbf{f}^k$  a  $\mathbf{J}^k$  stanovená pro konkrétní vektor  $\mathbf{x}^k$ , takže aproximace  $f(\mathbf{p})$  je platná pouze v určité okolí  $\mathbf{x}^k$ . Za touto hranicí je nahrazení funkce již nepřesné. To stejné samozřejmě platí také pro cílovou funkci  $m^k(\mathbf{p})$ . Celá situace je názorně zobrazena na obr. 5.2. Z tohoto obrázku je zřejmé, že krok  $\mathbf{p}$  při užití oblasti důvěryhodnosti se blíží řešení daleko lépe než bez ní.



Obrázek 5.2: Grafické znázornění principu Trust-Region metod

Nyní můžeme shrnout princip TR metod. Jedna iterace TR metod je optimalizační úloha, kdy cílem je nalézt krok  $\mathbf{p}$  tak, aby minimalizoval cílovou funkci  $m^k(\mathbf{p})$  definovanou v 5.41 a současně aby tento krok  $\mathbf{p}$  ležel uvnitř oblasti důvěryhodnosti definované poloměrem  $\Delta^k$ ; matematicky vyjádřeno

$$\mathbf{p}: \min m^k(\mathbf{p}), \text{ přičemž musí platit } |\mathbf{p}| < \Delta^k. \quad (5.42)$$

Metody TR jsou přímo dodávány v prostředí MATLAB a proto nebyly autorem programovány. MATLAB nabízí dvě varianty těchto metod, a to TR Reflective<sup>13</sup> a TR Dogleg. Obě metody jsou založeny na výše uvedeném principu, přičemž Dogleg metoda by měla být univerzálnější a Reflective metoda je určena pro rozsáhlé úlohy o velkém  $n$ . Podklady pro algoritmy metod lze nalézt například v [15] nebo [17].

## Zhodnocení Trust-Region metod

Trust-Region metody jsou známé svoji robustností, což znamená, že jsou vhodné i pro špatně podmíněné úlohy. Na testovací soustavách nelineárních rovnic uvedených v kap. 5.8 se však ukázaly TR metody jako málo efektivní ve srovnání s ostatními metodami.

## 5.5 Levenberg-Marquardt metoda

Metoda je kombinací Gauss-Newtonovy a gradientní metody. Tato metoda je taktéž dodávána přímo v prostředí MATLAB a proto nebyla autorem programována. Vzhledem k tomu, že metoda pro jedno ze dvou testovaných soustav rovnic vykazovala divergenci, není podrobně ji zkoumána ani popisována, viz obr. 5.5. Pro popis principu a algoritmu Levenberg-Marquardt metody viz [31] a [32].

<sup>13</sup> TR Reflective využívá jiný tvar cílové funkce  $m^k(\mathbf{p})$ , než je uvedeno v 5.41.

## 5.6 Genetické algoritmy

V mnoha zdrojích (například [33]) je popsáno, že při řešení soustav nelineárních rovnic lze využít genetické algoritmy, jejichž postup je ve zkratce následující. Nejdříve je vymezena oblast přípustného řešení pro každou proměnou  $x_i$

$$\begin{aligned} x_1 &\in \langle a_1, b_1 \rangle, \\ &\vdots \\ x_n &\in \langle a_n, b_n \rangle. \end{aligned} \quad (5.43)$$

Tyto oblasti jsou poté naděleny na  $m+1$  bodů

$$\begin{aligned} x_1 &\in \langle a_1, a_1 + h_1, a_1 + 2h_1, \dots, b_1 \rangle, \\ &\vdots \\ x_n &\in \langle a_n, a_n + h_n, a_n + 2h_n, \dots, b_n \rangle, \end{aligned} \quad \text{kde } h_i = \frac{b_i - a_i}{m}, \quad (5.44)$$

čímž vzniká síť bodů. Z této sítě je vybráno náhodně  $k$  vektorů  $x$  neboli jedinců, kteří tvoří generaci. Odhad minimálního počtu jedinců  $k$  se liší, ale nejméně by jich mělo být alespoň

$$k \geq 3 \cdot n. \quad (5.45)$$

Pro tyto jedince se vyčíslí hodnota  $|f|$  a podle velikosti této hodnoty se jedinci seřadí. Jedinci se dále vybírají, kříží a mutují, čímž vzniká nová generace, jejíž počet jedinců je opět  $k$ . Pro novou generaci se opět vyčíslí hodnota  $|f|$  a pokračuje se stejně jako již bylo uvedeno. S přibývajícimi generacemi se jedinci začínají shlukovat kolem řešení soustavy nelineárních rovnic. Pokud má soustava více řešení, je i těchto shluků je více. Nejčastějším ukončovacím kritériem u genetických algoritmů je počet generací. Po dosažení daného počtu generací je vyhodnoceno řešení jako poloha shluku jedinců.

Z popisu principu genetických algoritmů je zřejmé, že výhoda genetických algoritmů spočívá v tom, že hledají všechny kořeny soustavy nelineárních rovnic na námi zvolené oblasti 5.43. Tato vlastnost zde však nepřináší výhodu ale spíše naopak. Při výpočtu úlohy polohy máme téměř vždy k dispozici blízky počáteční odhad a hledáme řešení právě nejbližší tomuto odhadu. Vzhledem k tomu, že pro jisté hodnoty souřadnic mechanismu existuje několik variant poloh celé soustavy, hledáme pouze to nejbližší řešení vzhledem k předchozímu, aby nedocházelo k přeskokování do různých poloh některých členů. Proto je prohledávání celé oblasti nejen neefektivní vzhledem k počtu vyčíslení, ale taktéž zcestné pro řešení úlohy polohy mechanismů. Využití genetických algoritmů pro výpočet soustav nelineárních rovnic, které reprezentují polohu mechanismu, shledává autor jako zcela nevhodné.

## 5.7 Podmíněnost řešení soustav nelineárních rovnic

Pokud se zabýváme nelineárními soustavami rovnic, je nutné stanovit, jaké podmínky jsou nutné pro to, aby výpočet konvergoval k řešení soustavy. Funkci soustavy rovnic  $f(x)$  opět aproximujeme pomocí Taylorova rozvoje prvního řádu

$$f(x+p) = f(x) + J(x) \cdot p + O(|p|^2), \quad (5.46)$$

kde  $J(x)$  je Jacobiho matice viz 5.11,  $O(|p|^2)$  je chybová funkce, která nahrazuje prvky vyššího řádu Taylorova rozvoje a  $p$  je Newtonův krok. Pokud vektor  $x+p$  bude řešením soustavy pak bude platit

$$f(x+p) = 0. \quad (5.47)$$

Zanedbáme funkci  $O(|p|^2)$  a dosazením z 5.46 do 5.47 získáme vztah pro Newtonův krok

$$p = -\left(J(x)\right)^{-1} \cdot f(x). \quad (5.48)$$

Jak jsme si mohli všimnout výše, všechny metody až na genetické algoritmy využívají tohoto vztahu. Ze vztahu 5.48 vyplývá, že pokud Jacobiho matice  $J(x)$  je singulární, tak krok  $p$  nejsme schopni vypočítat. Tuto úvahu lze rozvinout dále – pokud je matice  $J(x)$  blízka singulární matici, krok  $p$  lze stanovit, ale bude značně nepřesný. Z toho vyplývá, že potřebujeme stanovit nejlépe číselnou charakteristiku „regulérnosti“ matice. Dle [38] se tato charakteristika nazývá číslo podmíněnosti matice a vypočítá se jako součin Frobeniových norem

$$\kappa(J) = \|J\|_2 \cdot \|J^{-1}\|_2 \geq 1. \quad (5.49)$$

Čím je číslo podmíněnosti  $\kappa$  bližší jedné, tím je matice lépe podmíněna a tím bude vypočítaný Newtonův krok přesnější. Samozřejmě čím je  $\kappa$  větší, tím je matice hůře podmíněna atd. Číslo podmíněnosti lze vypočítat pomocí numerických metod viz [35] bez sestavování inverzní matice, což je z pohledu efektivity drahá numerická operace.

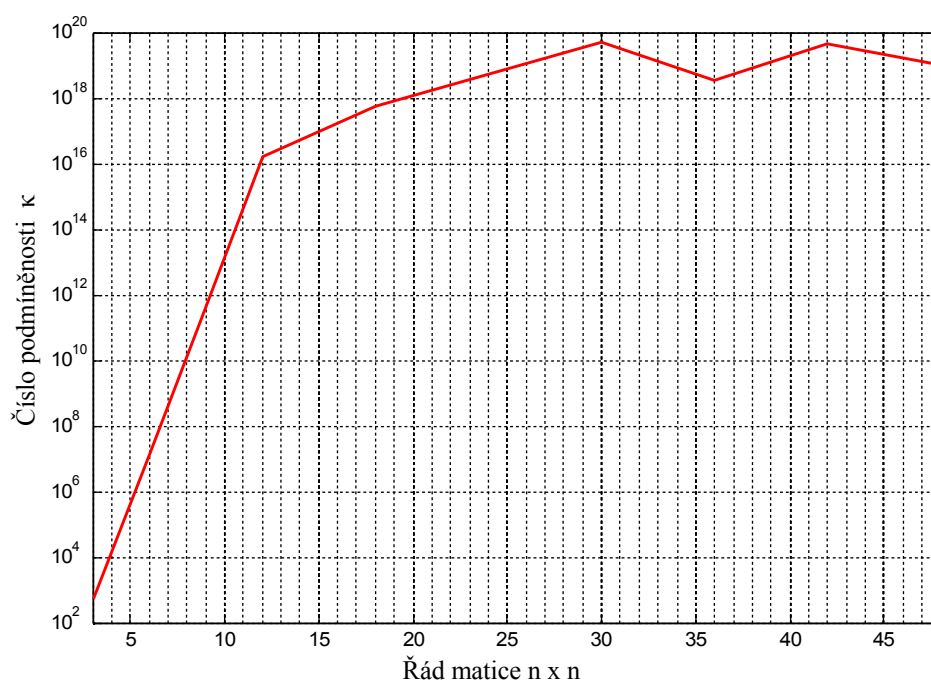
Pozn.: Frobeniova norma matice se stanoví jako

$$\|J\|_2 = \left( \sum_{i=1}^n \sum_{j=1}^n J_{ij}^2 \right)^{\frac{1}{2}}. \quad (5.50)$$

Vzhledem k tomu, že vztah 5.49 je v podstatě empirický, nelze najít přesnou hranici, kdy je matice ještě dobře podmíněná a kdy tomu již tak není. Proto uvedeme praktické příklady, které alespoň naznačí jaká čísla  $\kappa$  vypovídají o špatně podmíněné matici, což je například známá Hilbertova matice, jejíž tvar lze najít např. v [38]. V tabulce 5.1 jsou uvedena čísla podmíněnosti  $\kappa$  Hilbertovy matice v závislosti na řádu matice  $n$ . Tyto vypočtené hodnoty budeme označovat jako kritické  $\kappa_{krit}$ . Pro lepší představu je v levém sloupci uvedeno jakým typům mechanismů by odpovídala Jacobiho matice o tomto řádu  $n$ . V obr. 5.3 je uvedena tato závislost graficky.

Tabulka 5.1: Čísla podmíněnosti  $\kappa$  Hilbertovy matice v závislosti na řádu matice  $n$ 

	Řád matice $n \times n$ - Počet rovnic $n$	Číslo podmíněnosti $\kappa_{krit}$
Rovina – 1 smyčka	3	$5.2 \cdot 10^2$
Rovina – 2 smyčky nebo Prostor – 1 smyčka	6	$1.5 \cdot 10^7$
Rovina – 3 smyčky	9	$4.9 \cdot 10^{11}$
Rovina – 4 smyčky nebo Prostor – 2 smyčky	12	$1.7 \cdot 10^{16}$
Prostor – 3 smyčky	18	$6.3 \cdot 10^{17}$
Prostor – 4 smyčky	24	$5.6 \cdot 10^{18}$
Prostor – 5 smyček	30	$5.4 \cdot 10^{19}$
Prostor – 6 smyček	36	$3.8 \cdot 10^{18}$
Prostor – 7 smyček	42	$4.9 \cdot 10^{19}$
Prostor – 8 smyček	48	$1.1 \cdot 10^{19}$

Obrázek 5.3: Čísla podmíněnosti  $\kappa$  Hilbertovy matice v závislosti na řádu matice  $n$ 

## 5.8 Numerické testy metod pro řešení soustav nelineárních rovnic

Výše prezentované metody řešení soustav nelineárních rovnic potažmo jejich algoritmy byly aplikovány na testovací soustavy rovnic, které reprezentují reálné mechanismy. Ukončovacím kritériem u všech algoritmů byla velikost normy vektoru reziduálů  $\varepsilon_{NS}$ . Hodnota přesnosti  $\varepsilon_{NS}$  byla nastavena

$$\varepsilon_{NS} = 10^{-4} \text{ mm}. \quad (5.51)$$

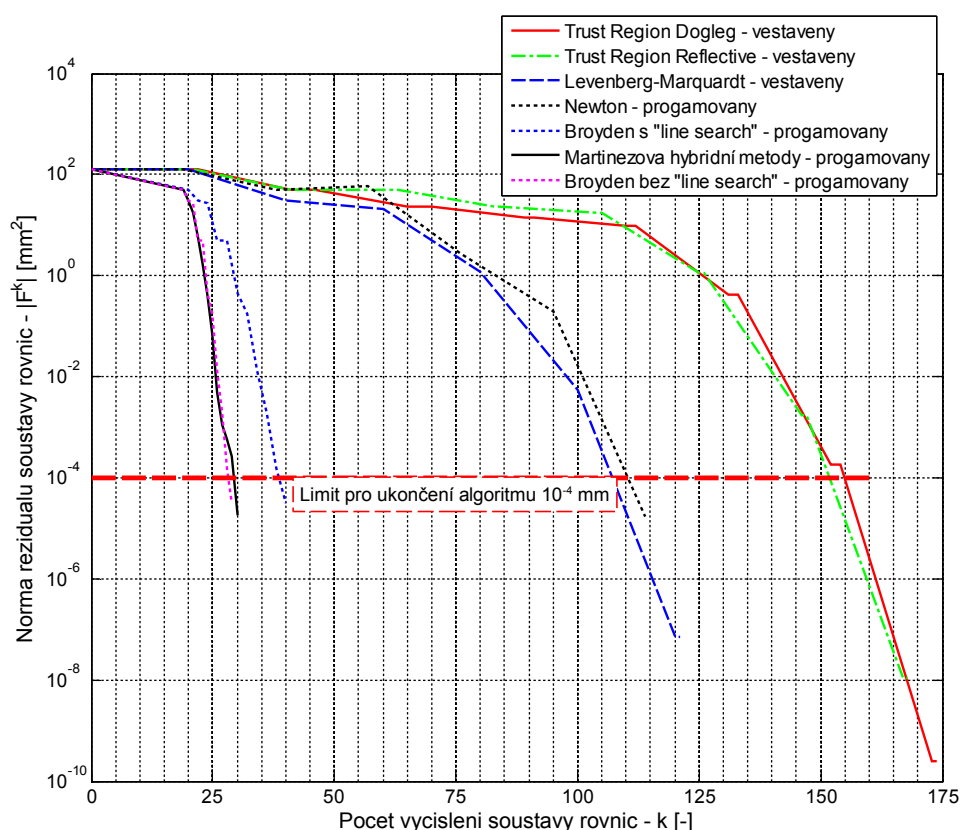
### 5.8.1 Testovací soustava A - 18 rovnic

První testovací soustava 18 rovnic, popisuje úlohu polohy mechanismu prezentovaného v kap 7.

První zajímavou charakteristikou soustavy je vzhledem k předchozí kapitole číslo podmíněnosti  $\kappa$  Jacobiho matice. Číslo  $\kappa$  bylo vyhodnoceno pro matici  $\mathbf{J}$  pro počáteční odhad vektoru  $\mathbf{x}$  a pro výsledný vektor  $\mathbf{x}_{min}$ , viz tab. 5.2. Všimněme si, že se hodnota výrazně nemění, což znamená, že podmíněnost matice  $\mathbf{J}$  je během výpočtu téměř neměnná. Dalším poznatkem je, že pokud vypočtenou hodnotu  $\kappa$  srovnáme s kritickou hodnotou  $\kappa_{krit}$  z tab. 5.1, lze prohlásit, že  $\kappa$  pro testovací soustavu A je o 13 řádů nižší, což jasně vypovídá o dobré podmíněnosti soustavy rovnic A.

Tabulka 5.2: Číslo podmíněnosti Jacobiho matice na počátku konci výpočtu

	Číslo podmíněnosti $\kappa$	Kritické číslo podmíněnosti $\kappa_{krit}$ pro $n = 18$ z tab. 5.1
Počáteční odhad vektoru $\mathbf{x}$	$3.17 \cdot 10^4$	$6.3 \cdot 10^{17}$
Výsledný vektor $\mathbf{x}$	$3.43 \cdot 10^4$	



Obrázek 5.4: Průběh konvergence jednotlivých metod v závislosti na počtu vyčíslení funkce  $f$

Průběh konvergence jednotlivých metod v závislosti na počtu vyčíslení funkce  $f$ , který je značen  $k$ , je vykreslen v obr. 5.4. Výsledné počty vyčíslení potřebné k dosažení přesnosti  $\varepsilon_{NS}$  jsou

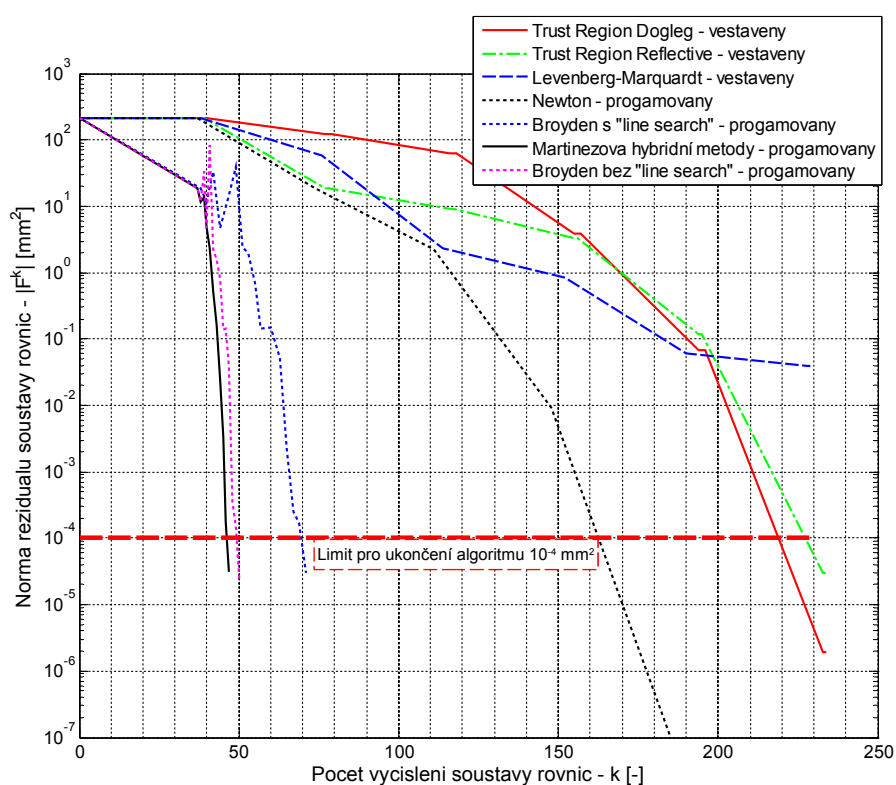


uvedeny v tab. 5.3, kde nejefektivnějšími metodami jsou Broydenova a Martinezova hybridní metoda, které dosáhly výsledného vektoru  $x$  během pouhých 29 respektive 30 vyčíslení  $f$ . Zajímavé je, že 18, to je 60 % všech vyčíslení, připadá na sestavení počátečního odhadu Jacobiho matice. Naopak nejhůře v testu efektivity dopadly trust-region metody, které potřebovaly k dosažení výsledku kolem 170-ti vyčíslení.

Tabulka 5.3: Srovnání jednotlivých metod dle počtu vyčíslení a konvergence

Nelineární soustava 18 rovnic o 18 neznámých reprezentující mechanismus z kap. 7			
	$k$ -počet vyčíslení funkce $f$	Hodnota $ f  \cdot 10^4$	
Trust-Region Dogleg	174	0.00	
Trust-Region Reflective	168	0.00	
Levenberg-Marquardt	121	0.00	
Newton	114	0.17	
Broyden s "line search"	40	0.31	
Martinez	30	0.18	Nejefektivnější
Broyden bez "line search"	29	0.31	Nejefektivnější

## 5.8.2 Testovací soustava B - 38 rovnic



Obrázek 5.5: Průběh konvergence jednotlivých metod v závislosti na počtu vyčíslení funkce  $f$

Druhá testovací soustava 36 rovnic, reprezentuje úlohu polohy mechanismu popsaného v kap 8.

Opět bylo vyhodnoceno číslo podmíněnosti Jacobiho matice pro počáteční odhad a pro výsledný vektor  $x$ . Opět lze odpozorovat, že podmíněnost matice  $J$  se nezměnila během výpočtu a že soustava je dobře podmíněna, protože hodnota  $\kappa$  je vzhledem ke kritické hodnotě o 11 řádů nižší.

Tabulka 5.4: Číslo podmíněnosti Jacobiho matice na počátku konci výpočtu

	Číslo podmíněnosti $\kappa$	Kritické číslo podmíněnosti $\kappa_{krit}$ pro $n = 36$ z tab. 5.1
Počáteční odhad vektoru $x$	$1.33 \cdot 10^7$	$3.8 \cdot 10^{18}$
Výsledný vektor $x$	$1.33 \cdot 10^7$	

Průběh konvergence jednotlivých metod v závislosti na počtu vyčíslení funkce  $f$  je vykreslen v obr. 5.5. Výsledné počty vyčíslení  $f$  potřebné k dosažení přesnosti  $\varepsilon_{NS}$  jsou uvedeny v tab. 5.5 kde nejefektivnějšími metodami jsou opět Martinezova hybridní a Broydenova metoda, které dosáhly výsledného vektoru  $x$  během pouhých 47 a 50 vyčíslení  $f$ . Nejhorší efektivitu vykazovaly opět trust-region metody, které potřebovaly k dosažení výsledku 234 vyčíslení. Metoda Levenberg-Marquardt dokonce divergovala.

Tabulka 5.5: rovnání jednotlivých metod dle počtů vyčíslení a konvergence

Nelineární soustava 36 rovnic o 36 neznámých reprezentující mechanismus z kap 8			
	$k$ - počet vyčíslení funkce $f$	Hodnota $ f  \cdot 10^4$	
Levenberg-Marquardt	229	389.88	Úloha divergovala
Trust-Region Dogleg	234	0.02	
Trust-Region Reflective	234	0.31	
Newton	185	0.00	
Broyden s "linesearch"	71	0.30	
Broyden bez "linesearch"	50	0.24	Nejefektivnější
Martinez	47	0.31	Nejefektivnější

### 5.8.3 Testovací soustava C - 2 rovnice

Pro úplnost byla třetí testovací soustava 2 rovnic vybrána z [33], kde je tato soustava doporučena jako jedna z obtížně řešitelných a proto vhodných soustav pro testování algoritmů jednotlivých metod. Nutno poznamenat, že tato soustava rovnic nereprezentuje žádný mechanismus. Testovací soustava C

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} e^{x_1} + x_1 \cdot x_2 - 1 \\ \sin(x_1 \cdot x_2) + x_1 + x_2 - 1 \end{pmatrix} = \mathbf{0}, \quad (5.52)$$

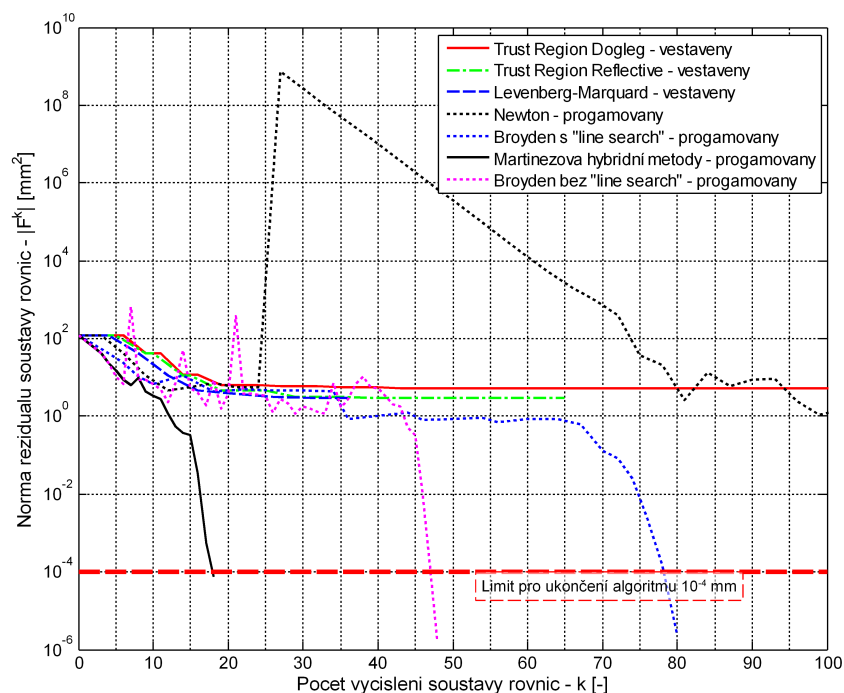
má kořeny

$$\mathbf{x}_{root} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \quad (5.53)$$

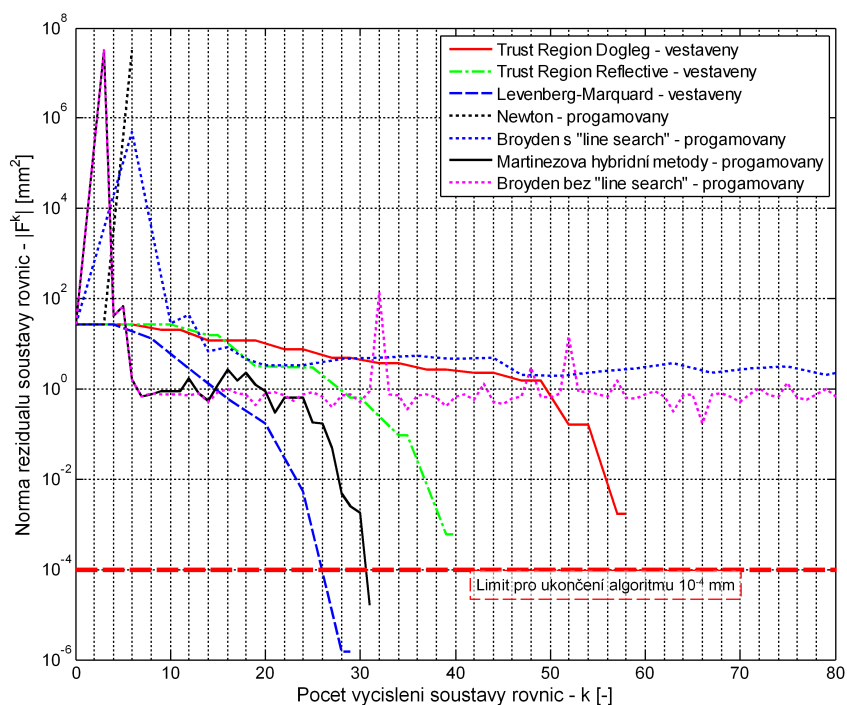
Test algoritmů byl proveden pro dva počáteční odhady

$$\mathbf{x}_I^0 = \begin{pmatrix} 5 \\ -5 \end{pmatrix} \quad \text{a} \quad \mathbf{x}_{II}^0 = \begin{pmatrix} -5 \\ -5 \end{pmatrix}. \quad (5.54)$$

Grafy konvergence jsou zobrazeny na obr. 5.6 a 5.7. Z grafů je vidět, že počáteční odhad řešení má rozhodující vliv nejen na efektivitu ale i na to, zda algoritmus konverguje či ne. V obou případech konvergovaly k řešení pouze Levenberg-Marquardt a Martinezova hybridní metoda. Efektivita obou metod byla střídavá.

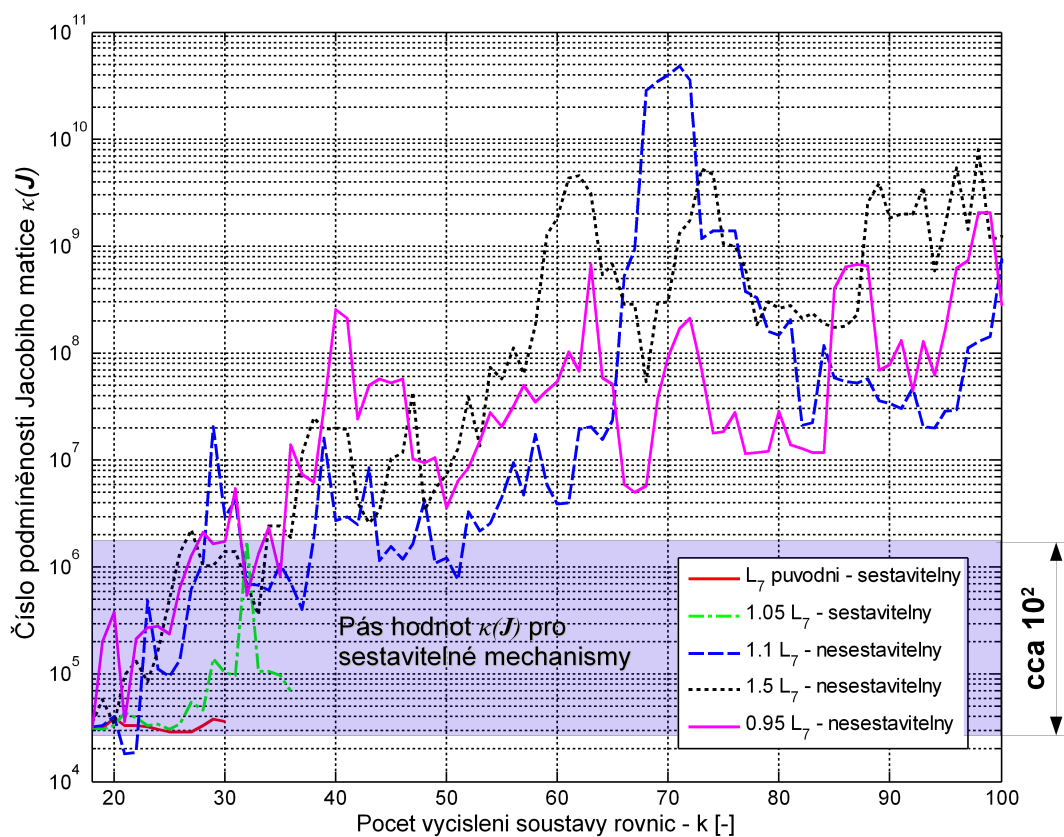


Obrázek 5.6: Průběh konvergence jednotlivých metod v závislosti na počtu vyčíslení funkce  $f$ ; počáteční odhad  $[5, -5]$



Obrázek 5.7: Průběh konvergence jednotlivých metod v závislosti na počtu vyčíslení funkce  $f$ ; počáteční odhad  $[-5, -5]$

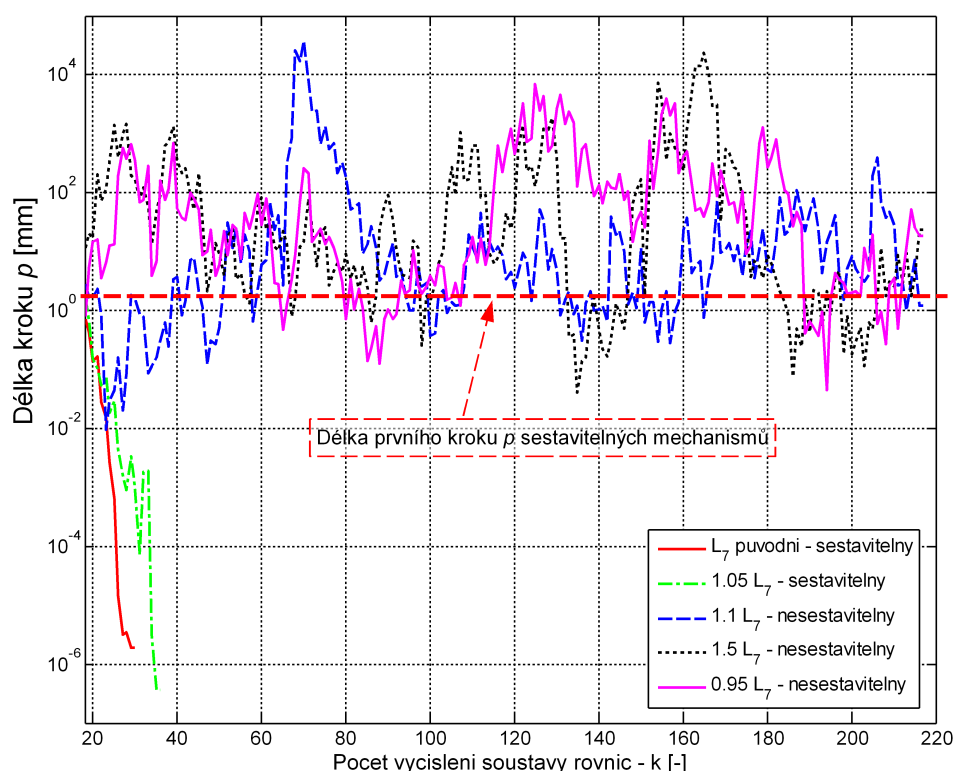
#### 5.8.4 Testovací soustava D – nesestavitelný mechanismus



Obrázek 5.8: Závislosti čísla podmíněnosti  $\kappa$  Jacobiho matice na počtu vyčíslení funkce  $f$

Při optimalizaci mechanismů je častým úkazem, že optimalizační algoritmus navrhne rozměry mechanismu tak, že soustavu nelze sestavit, detailněji viz kap. 4.6. Proto je zde věnována pozornost chování soustav nelineárních rovnic, které popisují polohu mechanismu, jež nelze z rozměrových důvodů sestavit. Pozorování bylo provedeno pro Martinezovu hybridní metodu, která se ukázala v numerických testech nejefektivnější.

Soustava nelineárních rovnic, která prezentuje nesestavitelný mechanismus, byla vytvořena z úlohy polohy popsané v kap. 7 změnou délky tyče  $L_7$ . Na obr. 5.8 vidíme závislosti čísla podmíněnosti  $\kappa$  Jacobiho matice na počtu vyčíslení funkce  $f$ . Závislosti jsou uvedeny pro několik modifikací délky  $L_7$  – v legendě je poznačeno, které modifikace délky umožňují sestavit mechanismus a které ne. Z grafu lze vypožorovat, že pro sestavitelné mechanismy se hodnoty podmíněnosti  $\kappa$  pohybují v pásu širokém přibližně dva řády. Hodnoty  $\kappa$  pro nesestavitelné mechanismy značně oscilují a postupně stoupají.



Obrázek 5.9: Závislosti délky kroku  $p$  na počtu vyčíslení funkce  $f$

Na obr. 5.9 jsou zobrazeny závislosti délky kroku  $p$  na počtu vyčíslení funkce  $f$ . Toto je opět provedeno pro stejné modifikace délky tyče  $L_7$ . Zde je zcela zřejmý rozdíl mezi křivkami odpovídajícími sestavitelným a nesestavitelným mechanismům. U sestavitelných mechanismů má délka kroku  $p$  s přibývajícími iteracemi jasně klesající tendenci, a to tak, že první krok je vždy nejdelší. Naopak u nesestavitelných mechanismů délka kroku značně osciluje, a to většinou nad délkou prvního kroku. Dále si všimněme, že délka prvního kroku je u všech modifikací podobná –

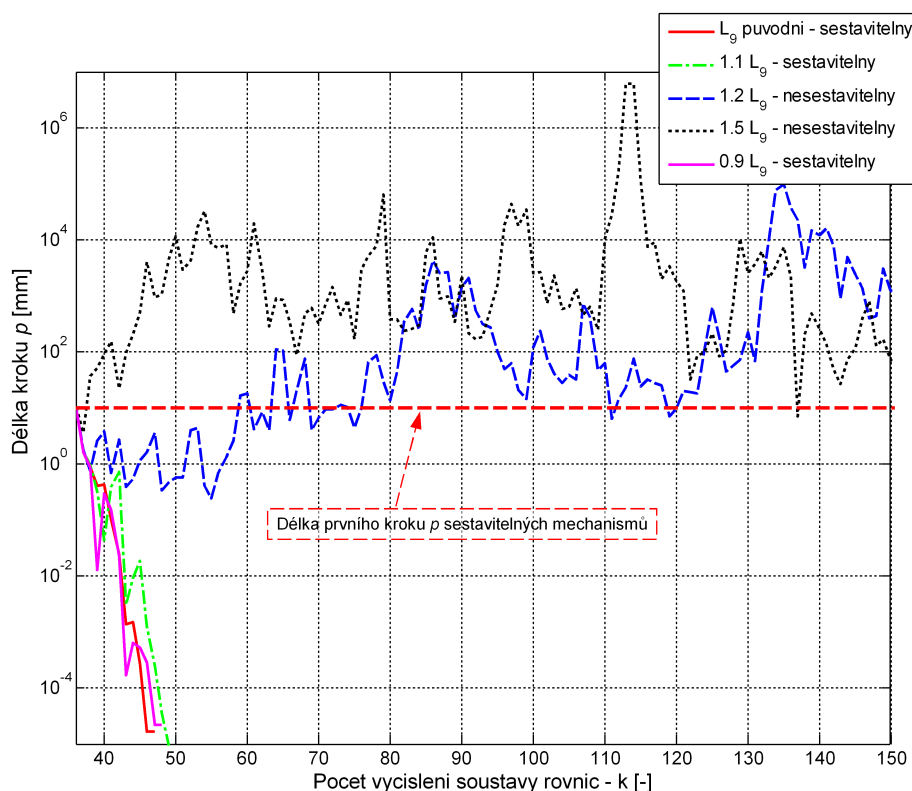
rozmezí jednoho řádu. Podstatné je, že stejné chování pro délku kroku  $p$  se potvrdilo u jiných mechanismů. Např. na obr. 5.10 je vyobrazen stejný typ závislosti pouze pro mechanismus z kap. 8, kde byla modifikována délka tyče  $L_9$ . Z výše uvedených poznatků plyne důležitý závěr, že do algoritmu pro výpočet soustav nelineárních rovnic, které reprezentují polohu mechanismu, je vhodné implementovat další ukončovací kritérium, které algoritmus ukončí a označí ho jako nesestavitelný. Kritérium je navrženo následovně

$$\text{pokud } |p^{k+1}| > tol \cdot |p^0| \text{ je mechanismus nesestavitelný a ukončí výpočet,} \quad (5.55)$$

kde  $tol$  představuje poměrnou toleranci vůči délce prvního kroku. Navrhujeme

$$tol > 10, \quad (5.56)$$

což znamená, že pokud krok  $p^{k+1}$  bude desetkrát delší než počáteční krok  $p^0$ , bude výpočet ukončen. To by v praxi znamenalo, že výpočet pro mechanismus zkoumaný v obr. 5.9 s modifikací  $0.95 L_7$  by byl ukončen již po druhé iteraci, což ušetří až stovky zbytečných vyčíslení  $f$ . Toto kritérium značně zefektivní optimalizační algoritmy.



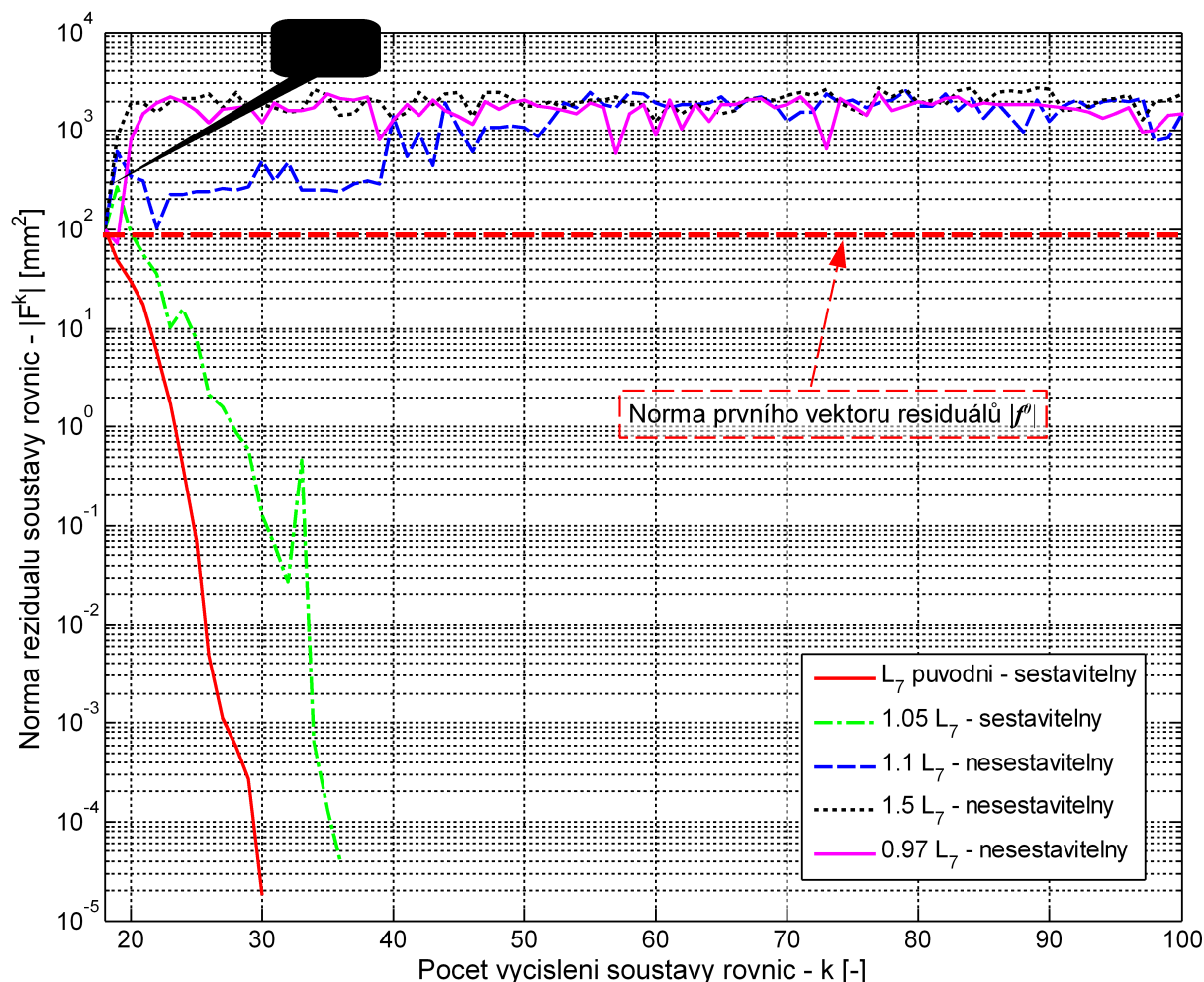
Obrázek 5.10: Závislosti délky kroku  $p$  na počtu vyčíslení funkce  $f$

V odborné literatuře se často objevuje jiné doplňkové ukončovací kritérium vztažené k délce vektoru residuálů, a to v podobě

$$\text{pokud } |f^{k+1}| > div \text{ je soustava rovnic neřešitelná - ukončí výpočet,} \quad (5.57)$$

kde je obtížné stanovit hodnotu čísla  $div$ . Stejná úvaha jako byla uvedena výše, a to vztáhnout

hodnotu  $\text{div}$  k počáteční hodnotě  $|\mathbf{f}^0|$ , se zde ukazuje jako nevhodná, protože kupříkladu v obr. Chyba: zdroj odkazu nenalezen se pro sestavitelný mechanismus hodnota  $|\mathbf{f}^k|$  výrazně zvyšuje vzhledem k normě residuálů na počátku  $|\mathbf{f}^0|$ . Proto toto pravidlo shledáváme jako nepoužitelné.



Obrázek 5.11: Důkaz neplatnosti kritéria navrhovaného v literatuře pro oblast mechanismů

## 5.9 Srovnání metod pro řešení soustav nelineárních rovnic

V této kapitole bylo popsáno šest metod pro řešení nelineárních soustav rovnic. U každé metody je uvedeno zhodnocení metody, zda je vhodná pro využití v zkoumané problematice, což je řešení úlohy polohy, která je prezentována soustavou nelineárních rovnic.

V kap. 5.8 byly provedeny tři numerické testy na modelových soustavách A, B, C a D. V tab. 5.6 jsou shrnuty výsledky těchto testů. Pro vyhodnocení uijeme dvě kritéria, a to spolehlivost a efektivitu. Z tabulky je zřejmé, že nejspolehlivější metodou je Martinezova hybridní metoda, která konvergovala k řešení ve všech případech. Z pohledu efektivity jsou jasně nejlepší dvě metody, a to Broydenova dobrá metoda a Martinezova hybridní metoda. Z tohoto vyplývá, že vzhledem ke spolehlivosti i efektivitě je nejvhodnější metodou pro řešení soustav nelineárních



rovníc reprezentujících polohu mechanismů Martinezova hybridní metoda. Je velice důležité připomenout, že na základě provedených testů (viz obr. 5.1) je zapotřebí u této metody na sestavit počáteční odhad Jacobiho matice pomocí vztahů pro numerický výpočet derivace dle 5.13. Použit jako počáteční odhad jednotkovou matici se ukázalo jako kriticky nevhodné – docházelo k divergenci algoritmu.

Tabulka 5.6: Shrnutí výsledků testů řešičů nelineárních soustav rovnic

Metoda	Testovací úloha A <i>k</i> - počet vyčíslení funkce <i>f</i>	Testovací úloha B <i>k</i> -počet vyčíslení funkce <i>f</i>	Testovací úloha C Konvergence <sup>14</sup>
Trust-Region Dogleg	174	234	D
Trust-Region Reflective	168	234	D
Levenberg-Marquardt	121	229 - D	K
Newton	114	185	D
Broyden s "linesearch"	40	71	D
Martinez	30	50	K
Broyden bez "linesearch"	29	47	D

Pro představu porovnejme efektivitu námi zvolené Martinezovy hybridní metody s přednastavenou metodou ve světově uznávaném a užívaném matematickém softwaru MATLAB 7.9.0, který nabízí tři metody - Trust-Region Dogleg, Trust-Region Reflective a metodu Levenberg-Marguardt, ze kterých je jako nejobecnější doporučována autory softwaru první metoda. Pokud bychom uvažovali pouze jeden výpočet úlohy polohy, tak dle

$$\frac{k_{\text{Dogleg test-A}}}{k_{\text{Martinez test-A}}} = \frac{174}{30} = 5.8 \quad (5.58)$$

je Martinezova metoda 5.8-krát efektivnější. Vyšší efektivita zvláště nabývá na významu u optimalizačních úloh, kde je úloha polohy řešena běžně v počtech deseti tisíců.

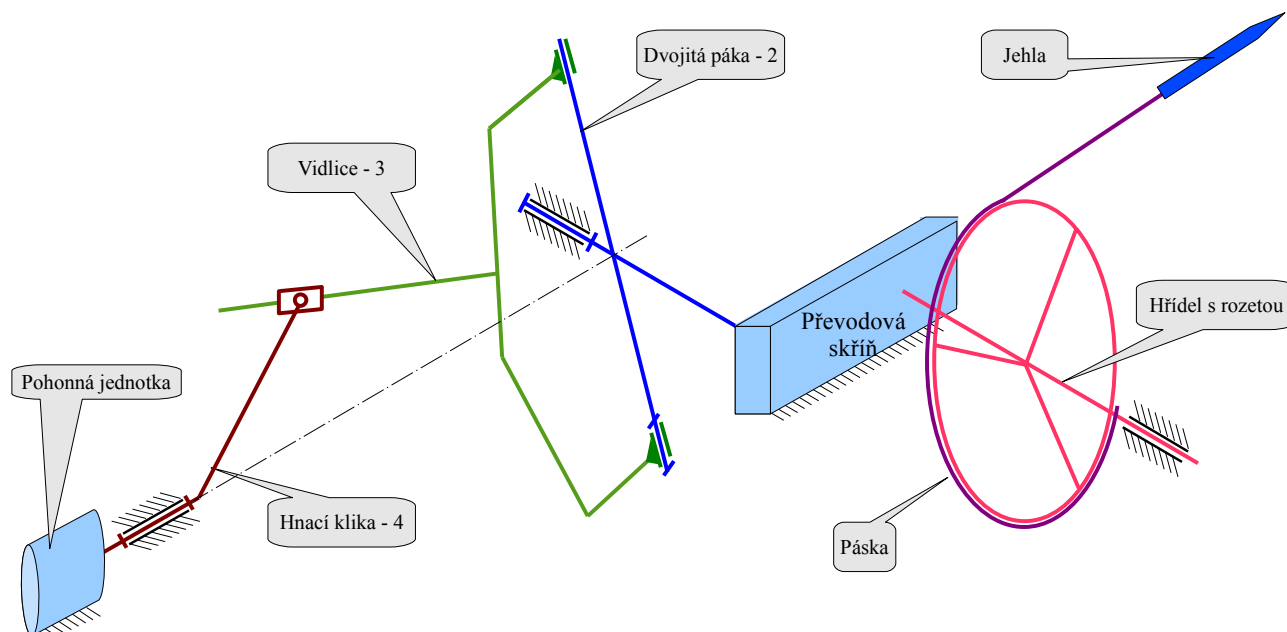
Dalším podstatným výstupem této kapitoly jsou poznatky o chování soustav rovnic, které popisují nesestavitelný mechanismus, viz kap. 5.8.4. Bylo zjištěno, že u sestavitelných mechanismů má délka kroku  $p$  s přibývajícími iteracemi jasně klesající tendenci, a to tak, že první krok je vždy nejdelší. Naopak u nesestavitelných mechanismů délka kroku  $p$  značně osciluje, a to většinou nad délkou prvního kroku, z čehož plyne důležitý závěr, že do algoritmu pro výpočet soustav nelineárních rovnic je vhodné implementovat další ukončovací kritérium 5.55. To v praxi znamená, že algoritmus je ukončen daleko dříve než došlo k vyčerpání maximálního počtu iterací, a proto je ušetřeno mnoho zbytečných vyčíslení funkce  $f$ . Toto kritérium je opět zvláště významné u

<sup>14</sup> K - úloha konvergovala, D - úloha divergovala.



optimalizace, kdy optimalizační algoritmus běžně navrhuje rozměry mechanismu tak, že soustavu nelze sestavit a pokud se tak stane, je potřeba co nejdříve ukončit algoritmus pro výpočet soustavy rovnic a přejít na speciální postup popsany v 4.6.

## 6 Prohozní mechanismus tkacího stroje – dynamická analýza



Obrázek 6.1: Schéma prohozovacího mechanismu

V této kapitole bude navržen postup řešení dynamické úlohy mechanismu při využití zjednodušeného výpočtu rychlostí a zrychlení dle kap. 3.3.10. Výsledky této analýzy budou srovnány s výsledky z multibody programového prostředí.

Prohozní mechanismus je součástí tkacího stroje. Tento mechanismus slouží k prohozování tkací jehly vedoucí tzv. útěk skrz podélnou soustavu nití, která se nazývá osnova, viz [36]. Schéma prohozovacího mechanismu je uvedeno na obr. 6.1.

Vstupním členem mechanismu je hnací klika 4, na kterou je připojena pohonná jednotka. Klika 4 konající rotační pohyb je kloubově-posuvně připojena k vidlici 3, která koná sférický pohyb. Vidlice 3 a dvojitá páka 2 jsou spojeny rotační vazbou. Rotace kliky 4 je přeměněna vidlicí na kyvný rotační pohyb páky 2. Kyvný pohyb páky 2 je přes převodovou skříň převedena na hřídel, na které je upevněna rozeta. Hřídel s rozetou koná také kyvný rotační pohyb pouze s jinými rychlostmi než páka 2. Na rozetě je namotána páska, která je přes vedení odmotávána a na konci této pásky je prohozovací jehla, v níž je příze osnovy.

Vzhledem ke snaze dosáhnout co nejrychlejšího procesu tkaní textilie musí být prohození jehly co nejrychlejší. Navíc páka 2, převodová skříň a rozeta konají kyvný pohyb, takže dochází ke změnám směrů vysokých rychlostí těchto členů a právě proto jsou velikosti zrychlení

jednotlivých členů značně vysoká a současně budou vysoké i dynamické síly.

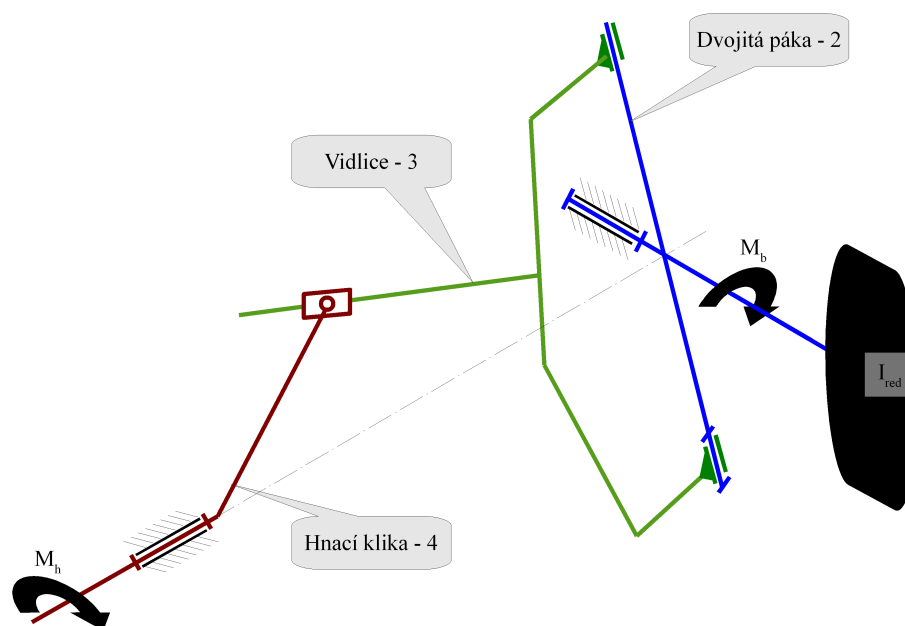
Z důvodu vysokých dynamických sil a rychlým změnám jejich směrů je aplikace vhodná pro testování metod řešení dynamických úloh.

## 6.1 Rozměry, zatížení a přijatá zjednodušení mechanismu

Při analýze prohozního mechanismu se omezíme pouze na tři členy, a to klik 4, vidlice 3 a páka 2. Pohonná jednotka mechanismu bude nahrazena hnacím konstantním momentem  $M_h$ . Převodová skříň, páka s rozetou a páska s jehlou budou nahrazeny momentem setrvačnosti  $I_{red}$  redukovaným na otáčky páky 2. Dále zavedeme pasivní odpory lineárně úměrné úhlové rychlosti, a to v podobě momentu  $M_b$  působícího na hřídel páky 2. Brzdný moment je vyjádřen funkcí

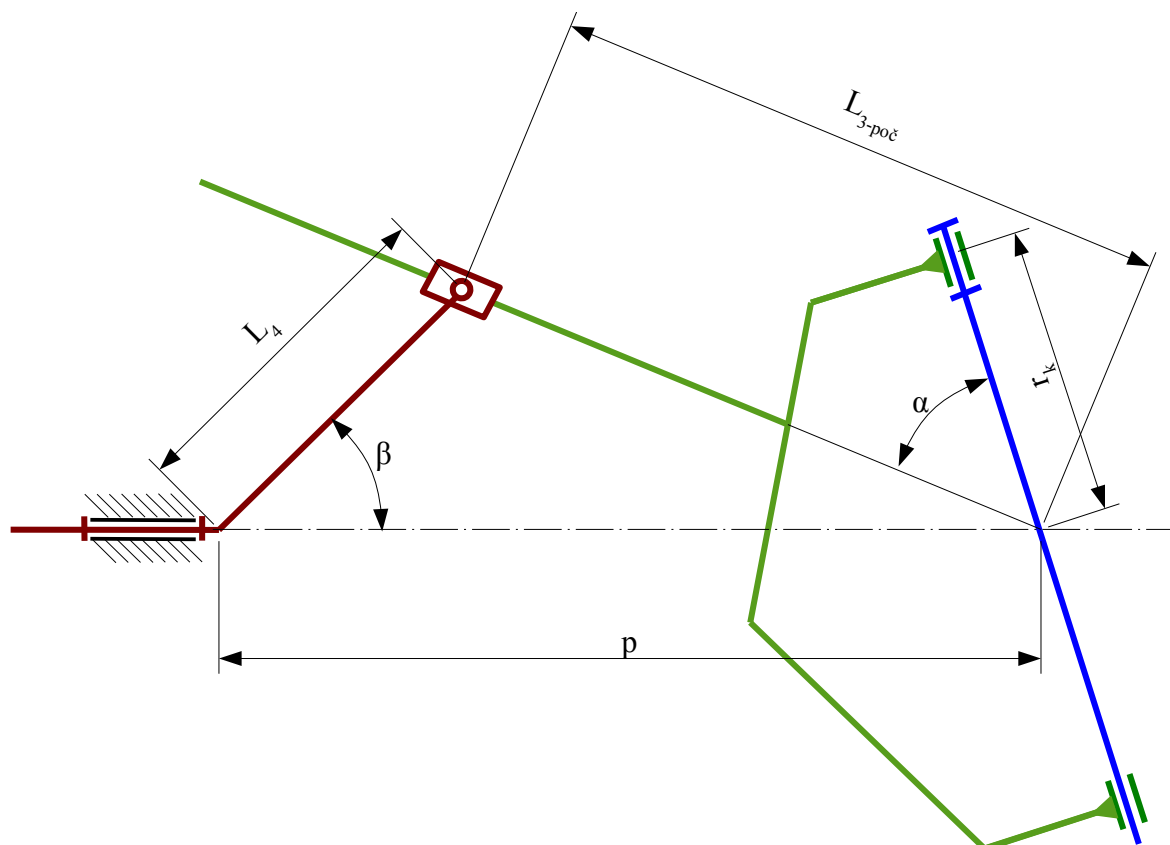
$$M_b = -b \cdot \omega_{páky2} \quad (6.1)$$

Schéma mechanismu se zjednoduší na obr. 6.2.



Obrázek 6.2: Zjednodušené schéma prohozovacího mechanismu

Rozměry mechanismu důležité pro kinematickou a dynamickou analýzu jsou vyznačeny v obr. 6.3. Mechanismus je zakreslen v čelním pohledu v poloze, kdy klika 4 a vidlice 3 leží v jedné rovině. Tuto polohu budeme dále považovat za počáteční polohu mechanismu. Dále je nutné upozornit, že střed sférického pohybu vidlice 3 leží na ose rotace hnací kliky 4, viz obr. 6.2. Proto je rozměr daný kótou polohy kloubové posuvné vazby  $L_{3poč}$  neměnný. Pokud by střed sférického pohybu ležel mimo tuto osu, rozměr  $L_{3poč}$  by byl proměnný. Úlohy polohy bude definovány v obecném tvaru a tudíž poloha kloubové posuvné vazby bude vnímána jako neznámá souřadnice vazby.



Obrázek 6.3: Rozměry mechanismu důležité pro kinematickou a dynamickou analýzu

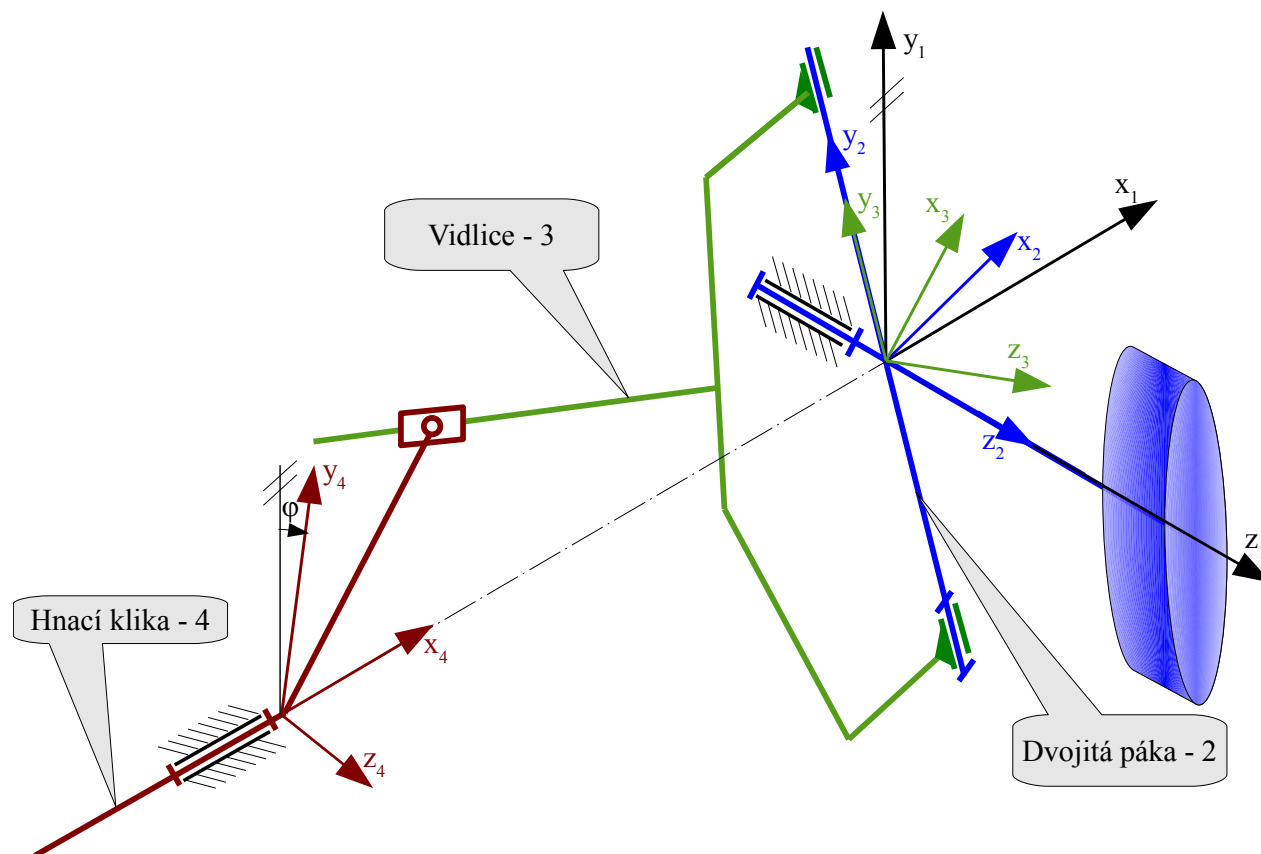
Číselné hodnoty jednotlivých technických parametrů jsou uvedeny v tab. 6.1.

Tabulka 6.1: Číselné hodnoty jednotlivých technických parametrů prohozního mechanismu

Rozměry			
$p$	540 mm	$\alpha$	$55^\circ$
$L_4$	434 mm	$\beta$	$48,9^\circ$
$L_{3poč}$	204 mm		
$r_k$	245 mm		
Zatížení a pasivní odpory			
$M_h$	100 N mm	$b$	1000 N mm s
Hmotnostní parametry (vztaženo k TSS daného tělesa, viz obr. 6.5)			
Dvojitá páka 2	$I_{2z} + I_{red}$	92,55 t mm <sup>2</sup>	
Hnací klika 4	$I_{4x}$	0,8966 t mm <sup>2</sup>	
Vidlice 3	$m_3$	6,715 10 <sup>-4</sup> t	
	$\boldsymbol{I}_3 = \begin{bmatrix} 24,823 & 15,159 & -0,001 \\ 15,159 & 27,107 & 0,001 \\ -0,001 & 0,001 & 51,922 \end{bmatrix} \text{ t mm}^2$		

## 6.2 Úloha polohy

Nejprve přiřadíme jednotlivým členům soustavy tělesové souřadné (dále TSS) systémy a také umístíme globální pevný souřadný systém (GSS), který bude označen indexem 1. Přiřazení jednotlivých souřadných systémů je provedeno na obr. 6.4. Připomeňme, že systémy musí být nutně pravotočivé a navíc tělesové souřadné systémy jsou pevně spojeny s jednotlivými tělesy.



Obrázek 6.4: Přiřazení tělesových souřadných systémů

Při výpočtu počtu stupňů volnosti musíme uvážit, že z kinematického hlediska je spojení vidlice 3 a páky 2 přeuročité - horní válcová vazba lze zcela vypustit a funkce mechanismu zůstane zachována. Přesto je toto spojení v praxi takto využíváno, a to z důvodu menšího namáhání vidlice 3 a páky 2. Při výpočtu stupňů volnosti myšleně odstraníme horní válcovou vazbu. Počet stupňů volnosti prohozního mechanismu dle 3.2 a s využitím tab. 3.1 bude

$$i = 6 \cdot (4 - 1) - (5 + 5 + 2 + 5) = 1. \quad (6.2)$$

Jedná se o mechanismus s jedním stupněm volnosti. Tento stupeň volnosti odpovídá jedné souřadnici mechanismu, což je natočení  $\varphi$  kliky 4, viz obr. 6.4. Všimněme si, že u tohoto mechanismu se nevyskytuje žádné těleso s tzv. volným stupněm volnosti, jak bylo uvedeno v kap. 3.2.3. Proto se počet stupňů volnosti  $i$  rovná počtu souřadnic mechanismu.

Počet základních smyček soustavy je roven jedné

$$r = 1. \quad (6.3)$$

Toto lze odvodit pomocí Postup tvorby základních smyček uvedeného v kap. 3.2.4. Mechanismus je tvořen jedním uzavřeným kinematickým řetězcem a žádnými dalšími otevřenými jednoduchými řetězci, a proto je zde pouze jedna základní smyčka. Připomeňme, že místo dvou vazeb mezi vidlicí 3 a pákou 2 stále uvažujeme pouze dolní rotační vazbu.

Vzhledem k tomu, že má mechanismus jednu základní smyčku, bude úloha polohy definována jednou maticovou rovnicí dle vzoru 3.44. Maticová rovnice prohozního mechanismu je následující

$$\mathbf{E} = \mathbf{T}_{12} \cdot \mathbf{T}_{23} \cdot \mathbf{T}_{34} \cdot \mathbf{T}_{41}. \quad (6.4)$$

Rozepišme jednotlivé transformační matice přechodu mezi jednotlivými souřadnými systémy dle obr. 6.5, přičemž všechny neznámé souřadnice vazeb (natočení nebo posunutí) budou značeny písmenem  $q$ . Matice  $\mathbf{T}_{12}$  značí přechod z GSS 1 do TSS 2. Abychom přemístili GSS 1 do TSS 2, stačí potočit GSS 1 kolem osy  $z_1$  o neznámý úhel  $q_1$ , takže transformační matice  $\mathbf{T}_{12}$  je zapsána pomocí transformačních matic základních pohybů

$$\mathbf{T}_{12} = \mathbf{R}_z(q_1). \quad (6.5)$$

Z TSS 2 do TSS 3 přejdeme natočením TSS 2 kolem osy  $y_2$  o neznámý úhel  $q_2$ , takže

$$\mathbf{T}_{23} = \mathbf{R}_y(q_2). \quad (6.6)$$

Abychom získali matici  $\mathbf{T}_{34}$ , natočíme TSS 3 kolem osy  $z_3$  tak, aby osa  $x$  byla shodná s ramenem, na které je připojena klika 4. Tomuto natočení odpovídá záporně vzatý úhel  $90^\circ - \alpha$ , viz obr. 6.4. Nyní natočený souřadný systém posuneme ve směru osy  $x$  do místa kloubově posuvné vazby. Toto posunutí je složeno ze dvou posunutí. Nejprve posuneme o délku  $-L_{3poč}$  (záporné znaménko z důvodu posunutí v záporném směru  $x$ ) a poté o neznámou souřadnici vazby  $q_3$ , která na základě předchozí, uvedené úvahy by měla zůstat nulová. Vzhledem k tomu, že je ve vazbě kloub, musíme nyní třikrát natočit souřadný systém o neznámá natočení  $q_4$ ,  $q_5$  a  $q_6$ , a to tak, aby osa  $y$  souřadného systému byla shodná s ramenem kliky 4 a osa  $x$  ležela v rovině kliky 4. Nyní posuneme systém o  $L_4$  do počátku TSS 4 ve směru osy  $y$ . Natočením kolem osy  $z_4$  o úhel  $90^\circ - \beta$  se nalézáme v kýženém TSS 4. Matice  $\mathbf{T}_{34}$  je zapsána

$$\mathbf{T}_{34} = \mathbf{R}_z(-90^\circ + \alpha) \cdot \mathbf{T}_x(-L_{3poč}) \cdot \mathbf{T}_x(q_3) \cdot \mathbf{R}_x(q_4) \cdot \mathbf{R}_y(q_5) \cdot \mathbf{R}_z(q_6) \cdot \mathbf{T}_y(-L_4) \cdot \mathbf{R}_z(90^\circ - \beta). \quad (6.7)$$

Poslední transformační maticí z maticové rovnice 5.54 je  $\mathbf{T}_{41}$ , což odpovídá přesunutí TSS 4 zpět do GSS 1. Toto provedeme rotací TSS 4 kolem osy  $x_4$  o úhel  $\varphi$ , který musí být vzat

záporně, aby byl dodržen kladný směr definovaný v obr. 6.4. Poté tento otočený systém posuneme do počátku GSS 1, a to posunutím ve směru osy  $x$  o vzdálenost  $p$ , čímž získáváme celou matici

$$\mathbf{T}_{41} = \mathbf{R}_x(\varphi) \cdot \mathbf{T}_x(p). \quad (6.8)$$

Maticová rovnice 5.54 spolu s rozepsanými transformačními maticemi v 6.5, 6.6, 6.7 a 6.8 popisuje polohu prohozního mechanismu. Z této maticové rovnice je nezávislých pouze 6 skalárních rovnic, které lze vybrat dle schématu 3.45. Neznámými v těchto rovnicích jsou souřadnice jednotlivých vazeb. Souřadnicí mechanismu je natočení  $\varphi$  kliky 4. Tuto soustavu nelineárních rovnic lze efektivně řešit použitím navrhovaného algoritmu VII.

### 6.3 Rychlosti a zrychlení

Velikost dynamických účinků je závislá na následujících kinematických veličinách

$$\alpha_{4x,4}, \alpha_{2z,2}, \omega_{13,3}, \alpha_{13,3}. \quad (6.9)$$

Pokud bychom chtěli stanovit zrychlení z maticové rovnice dle 3.52, měla by tato rovnice 20 sčítanců, což znamená, že by byla značně rozsáhlá a proto budou rychlosti i zrychlení stanoveny dle kap. 3.3.10. Veličiny z 6.9 vyjádříme jako funkci převodů a rychlosti a zrychlení souřadnice mechanismu. Vstupní souřadnice prohozního mechanismu je natočení  $\varphi$  kliky 4, takže všechny potřebné rychlosti a zrychlení budou vypočteny z úhlové rychlosti a zrychlení kliky 4. Dle kap. 3.2.7 lze vyjádřit úhlové zrychlení páky 2

$$\alpha_{2z,2} = p_1 \cdot \alpha_{4x,4} + dp_1 \cdot \omega_{4x,4}^2, \text{ kde } \omega_{4x,4} = \dot{\varphi}, \text{ a } \alpha_{4x,4} = \ddot{\varphi}. \quad (6.10)$$

Vzhledem k tomu, že zrychlení páky 2  $\alpha_{2z,2}$  je totožné se zrychlením úhlu natočení rotační vazby, viz obr. 6.4, platí

$$\alpha_{2z,2} = \ddot{q}_1 \quad (6.11)$$

lze převod  $p_1$  a jeho derivaci  $dp_1$  stanovit ze souřadnice vazby  $q_1$

$$p_1 = \frac{dq_1}{d\varphi} \text{ a } dp_1 = \frac{d^2 q_1}{d\varphi^2}. \quad (6.12)$$

Hodnoty převodu a jeho derivace lze stanovit numerickou derivací dle kap. 3.3.10, takže například převod  $p_1$  bude pro nejnižší řád přesnosti roven

$$p_1 = \frac{dq_1}{d\varphi} = (-q_1(\varphi_0) + q_1(\varphi_0 + h)) \cdot \frac{1}{h}, \quad (6.13)$$

kde  $\varphi_0$  je hodnota vstupní souřadnice neboli poloha celého mechanismu, pro kterou stanovujeme hodnotu převodu  $p_1$ . Z tohoto vztahu vyplývá, že je nutné dvakrát vypočítat úlohu polohy

mechanismu, abychom získali hodnoty natočení rotační vazby  $q_1(\varphi_0)$  a  $q_1(\varphi_0+h)$  pro dvě hodnoty vstupní souřadnice  $\varphi_0$  a  $\varphi_0+h$ . Obdobně lze stanovit hodnotu  $dp_1$

$$dp_1 = \frac{d p_1}{d \varphi} = \frac{d^2 q_1}{d \varphi^2} = (q_1(\varphi_0-h) - 2 \cdot q_1(\varphi_0) + q_1(\varphi_0+h)) \cdot \frac{1}{h^2}. \quad (6.14)$$

Úhlovou rychlost a zrychlení vidlice 3 získáme opět převodovými funkcemi. Úhlové rychlosti stanovíme z matice rychlosti dle 3.24

$$V_{13} = T_{13}^{-1} \cdot P_{13} \cdot \dot{\varphi}, \text{ kde } \Omega_{13,3} = \text{submatice}[V_{13}]^{1-3 \times 1-3}, \quad (6.15)$$

kde  $P_{13}$  je převod transformační matice  $T_{13}$ , který stanovíme numerickou derivací

$$P_{13} = \frac{d T_{13}}{d \varphi} = (-T_{13}(\varphi_0) + T_{13}(\varphi_0+h)) \cdot \frac{1}{h}. \quad (6.16)$$

Úhlová zrychlení vidlice 3 budou

$$A_{13} = T_{13}^{-1} [P_{13} \cdot \ddot{\varphi} + (DP_{13} - P_{13} \cdot T_{13}^{-1} \cdot P_{13}) \dot{\varphi}^2], \text{ kde } \Omega_{13,3} = \text{submatice}[A_{13}]^{1-3 \times 1-3}, \quad (6.17)$$

kde matice derivace řevodu transformační rovnice stanovíme opět numerickou derivací

$$DP_{13} = \frac{d P_{13}}{d \varphi} = \frac{d^2 T_{13}}{d \varphi^2} = (T_{13}(\varphi_0-h) - 2 \cdot T_{13}(\varphi_0) + T_{13}(\varphi_0+h)) \cdot \frac{1}{h^2}. \quad (6.18)$$

## 6.4 Základní vztahy dynamiky mechanismů v maticovém vyjádření

Zavedme následující označení pro vektor síly a momentu silové dvojice

$$F_{Ai}^j = \begin{bmatrix} F_{ix}^j \\ F_{iy}^j \\ F_{iz}^j \end{bmatrix}, \quad M_{Ai}^j = \begin{bmatrix} M_{ix}^j \\ M_{iy}^j \\ M_{iz}^j \end{bmatrix}, \quad (6.19)$$

kde horní index  $j$  označuje člen mechanismu, na který veličina působí, dolní index  $i$  souřadný systém, ve kterém jsou vyjádřeny složky dané veličiny a písmeno  $A$  označuje u síly její působiště a u momentu působiště síly, od které vznikl. Složky obou veličin lze transformovat do jiného souřadného systému

$$F_{Ak}^j = S_{ki} \cdot F_{Ai}^j, \quad M_k^j = S_{ki} \cdot M_i^j + R_{kA} \cdot S_{ki} \cdot F_{Ai}^j, \quad (6.20)$$

kde  $S_{ki}$  je matice směrových kosinů definovaná v 3.13 a  $R_{ki}$  je průvodič působiště  $A$  síly  $F$  vyjádřený v souřadném systému  $k$ . Průvodič je zapsán antisymetrickou maticí

$$R_{kA} = \begin{bmatrix} 0 & -z_{kA} & y_{kA} \\ z_{kA} & 0 & -x_{kA} \\ -y_{kA} & x_{kA} & 0 \end{bmatrix}. \quad (6.21)$$



Nyní můžeme zavést souhrnný vektor

$$\mathbf{B}_i^j = \begin{bmatrix} \mathbf{F}_{Ai}^j \\ \mathbf{M}_i^j \end{bmatrix}, \quad (6.22)$$

který bude dále nazýván bivektor.

Při znalosti těchto základních vztahů lze definovat pohybovou rovnici jednoho členu mechanismu

$$\sum \mathbf{B}_k^{jA} + \mathbf{B}_k^{jD} + \sum \mathbf{B}_k^R = \mathbf{0}, \quad (6.23)$$

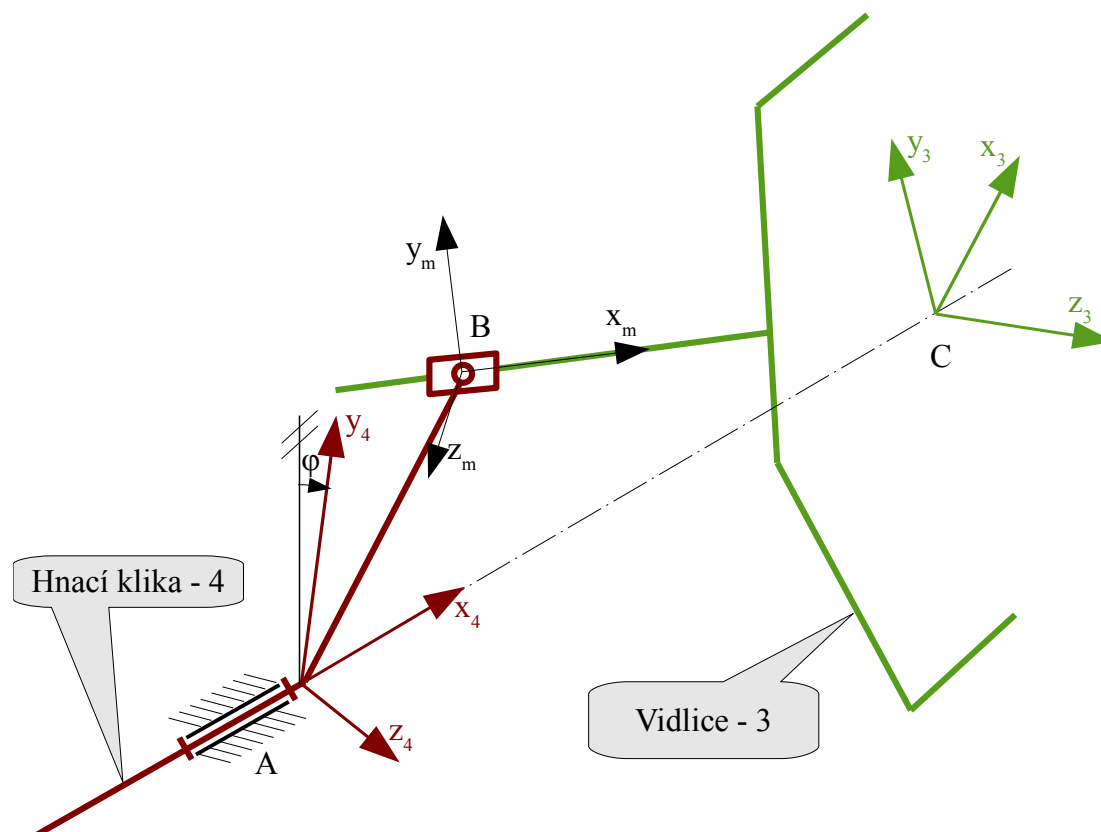
kde  $\mathbf{B}_k^{jA}$  je bivektor akčních sil působících na člen,  $\mathbf{B}_k^{jD}$  dynamické účinky a  $\mathbf{B}_k^{jR}$  je bivektor reakčních účinků.

Pro prohozní mechanismus budou z důvodu jednoduchosti dynamických účinků sestaveny pouze vlastní pohybové rovnice jednotlivých členů mechanismu dle tab. 6.2 takže na dynamické účinky a na reakce, které nemají vliv na pohyb mechanismu, nebude kladen důraz. Pohybové rovnice budou vždy sestavovány v TSS konkrétního tělesa.

Tabulka 6.2: Vlastní pohybové rovnice jednotlivých členů mechanismu

<i>Těleso</i>	<i>Druh pohybu</i>	<i>Pohybová rovnice</i>
<i>Hnací klika 4</i>	Rotace kolem $x_4$	$\sum M_{x4} = 0$
<i>Vidlice 3</i>	Sférický pohyb se středem v počátku TSS 3	$\sum M_{x3} = 0,$ $\sum M_{y3} = 0,$ $\sum M_{z3} = 0$
<i>Dvojitá páka 2</i>	Rotace kolem $z_2$	$\sum M_{z2} = 0$

## 6.5 Pohybové rovnice jednotlivých členů



Obrázek 6.5: Zavedení pomocného systému  $m$ , který je pevně spojen s vidlicí 3

### Hnací klika 4

Vlastní pohybovou rovnici kliky 4 získáme rovnicí

$$\sum M_{x_4} = 0, \quad (6.24)$$

z čehož vyplývá, že následovně budou brány na zřetel pouze ty veličiny, které mají momentový účinek kolem osy  $x_4$ .

Rotační vazbu  $A$  kliky 4 není třeba uvolňovat, protože nemá vliv na vlastní pohybovou rovnici. Akční veličinou působící na kliku 4 je pouze zatěžující moment  $M_h$

$$\mathbf{M}_{A4}^{4A} = \begin{bmatrix} M_h \\ 0 \\ 0 \end{bmatrix}, \quad (6.25)$$

Kloubová posuvná vazba  $B$  musí být nahrazena dvěma na sebe kolmými silami, které působí v rovině kolmé na rameno vidlice 3. Byl zaveden pomocný systém  $m$ , který je pevně spojen

s vidlicí 3, viz obr. 6.5. Neznámé složky reakcí vazby **B** lze vyjádřit v tomto systému

$$\mathbf{F}_{Bm}^{3R} = \begin{bmatrix} 0 \\ F_{Bmy}^{3R} \\ F_{Bmz}^{3R} \end{bmatrix}. \quad (6.26)$$

Pro pohybovou rovnici kliky 4 je potřeba použitím 6.20 transformovat tuto veličinu do TSS 4. Transformační matice lze odvodit jako matici inverzního pohybu k  $\mathbf{T}_{34}$  definovaném v 6.7

$$\mathbf{T}_{4m} = \mathbf{R}_z(-90^\circ + \beta) \cdot \mathbf{T}_y(L_4) \cdot \mathbf{R}_z(-q_6) \cdot \mathbf{R}_y(-q_5) \cdot \mathbf{R}_x(-q_4). \quad (6.27)$$

Matici  $\mathbf{S}_{4m}$  lze získat jako submatici  $\mathbf{T}_{4m}$  dle 3.13. Potom složky reakce vazby **B** lze transformovat do TSS 4 dle vztahu

$$\mathbf{F}_{B4}^{3R} = \mathbf{S}_{4m} \cdot \mathbf{F}_{Bm}^{3R} \quad (6.28)$$

Moment této reakce lze získat

$$\mathbf{M}_{B4}^{3R} = \mathbf{R}_{4B} \cdot \mathbf{F}_{B4}^{3R}, \quad (6.29)$$

kde průvodič působíště **B** lze získat z matice  $\mathbf{T}_{4m}$ , protože počátek systému *m* je shodný s působíštěm **B**, viz 3.12. Takže souřadnice působíště jsou

$$x_{4B} = T_{4m(14)}, y_{4B} = T_{4m(24)}, y_{4B} = T_{4m(34)}. \quad (6.30)$$

Dynamické účinky jsou vyjádřeny doplňkovým momentem pro rotační pohyb následně

$$\mathbf{M}_4^{4D} = \begin{bmatrix} -I_{4x} \cdot \alpha_{4x,4} \\ 0 \\ 0 \end{bmatrix}, \quad (6.31)$$

Vlastní pohybovou rovnici získáme dle 6.23 sečtením *x*-ových složek všech reakčních, akčních a doplňkových momentů

$$M_{A4x}^{4R} + M_{B4x}^{3R} + M_{4x}^{4D} = 0. \quad (6.32)$$

### Vidlice 3

Vidlice 3 koná sférický pohyb se středem v počátku TSS 3, takže vlastní pohybové rovnice získáme z

$$\begin{aligned} \sum M_{x3} &= 0, \\ \sum M_{y3} &= 0, \\ \sum M_{z3} &= 0. \end{aligned} \quad (6.33)$$

Nejprve transformujeme složky reakce ve vazbě **B** do TSS 3 dle

$$\mathbf{F}_{B3}^{3R} = -\mathbf{S}_{3m} \cdot \mathbf{F}_{Bm}^{3R}, \quad (6.34)$$

kde záporné znaménko vyjadřuje zákon akce a reakce a matice směrových kosinů  $\mathbf{S}_{3m}$  lze získat jako submatici transformační matice  $\mathbf{T}_{3m}$ , která je definována s přihlédnutím k 6.7 jako

$$\mathbf{T}_{3m} = \mathbf{R}_z(-90^\circ + \alpha) \cdot \mathbf{T}_x(-L_{3poc}) \cdot \mathbf{T}_x(q_3). \quad (6.35)$$

Moment reakce  $\mathbf{F}_{B3}^{3R}$  je k počátku TSS 3

$$\mathbf{M}_{B3}^{3R} = \mathbf{R}_{3B} \cdot \mathbf{F}_{B3}^{3R}, \quad (6.36)$$

kde průvodič  $\mathbf{R}_{3B}$  lze opět získat z posledního sloupce matice  $\mathbf{T}_{3M}$ , protože působíště reakce  $\mathbf{F}_{B3}^{3R}$  je shodné s počátkem souřadného systému  $m$ , takže prvky matice  $\mathbf{R}_{3B}$  jsou

$$x_{3B} = T_{3m\langle 14 \rangle}, y_{3B} = T_{3m\langle 24 \rangle}, y_{3B} = T_{3m\langle 34 \rangle}. \quad (6.37)$$

Jak již bylo popsáno výše, lze vazby mezi vidlicí 3 a pákou 2 nahradit jednou rotační vazbou. Představme si tuto vazbu ve středu sférického pohybu  $C$  nebo-li v počátku TSS 3, takže vzhledem k tvarům pohybových rovnic 6.33 stačí uvolnit tuto rotační vazbu pouze momenty ve směru os  $x_3$  a  $z_3$

$$\mathbf{M}_{C3}^{3R} = \begin{bmatrix} M_{3x}^{3R} \\ 0 \\ M_{3z}^{3R} \end{bmatrix}. \quad (6.38)$$

Doplňkové účinky sférického pohybu lze získat dle str. 44 z [37]

$$\mathbf{M}_3^{3D} = -(\mathbf{I}_3 \cdot \boldsymbol{\alpha}_{13,3} + \boldsymbol{\Omega}_{13,3} \cdot \mathbf{I}_3 \cdot \boldsymbol{\omega}_{13,3}). \quad (6.39)$$

Pro vidlici 3 pohybující se sférickým pohybem získáme dle 6.23 pohybové rovnice součtem momentů ve směru všech os TSS 3

$$\mathbf{M}_{B3}^{3R} + \mathbf{M}_{C3}^{3R} + \mathbf{M}_3^{3D} = \mathbf{0}. \quad (6.40)$$

## Dvojitá páka 2

Dle zákona akce a reakce bude vazba  $C$  uvolněna a transformována do TSS 2 dle vztahu

$$\mathbf{M}_{C2}^{3R} = -\mathbf{S}_{23} \cdot \mathbf{M}_{C3}^{3R}. \quad (6.41)$$

Rotační vazbu páky 2, jejíž osa je shodná s osou rotace tělesa, není třeba uvolňovat, protože nemá vliv na vlastní pohybovou rovnici. Doplnkový účinek rotačního pohybu páky 2 kolem osy  $z_2$  je vyjádřen momentem

$$\mathbf{M}_{C2}^{2D} = \begin{bmatrix} 0 \\ 0 \\ -I_{z2} \cdot \alpha_{2z,2} \end{bmatrix}. \quad (6.42)$$

Odpor prostředí je vydřen dle 6.1 momentem

$$M_{C2}^{2A} = \begin{bmatrix} 0 \\ 0 \\ -b \cdot \omega_{2z,2} \end{bmatrix}. \quad (6.43)$$

Pohybová rovnice dle 6.23 bude

$$M_{C2z}^{3R} + M_{C2z}^{2D} + M_{C2z}^{2A} = 0. \quad (6.44)$$

## 6.6 Algoritmy výpočtu dynamické úlohy

Navržený postup výpočtu dynamické úlohy se liší od tradičních postupů tím, že pro výpočet rychlostí a zrychlení nejsou využity rovnice definované v kap. 3.3.9, ale k jejich výpočtu jsou užity převodové funkce, které jsou určovány numericky. Z tohoto je již zřejmé, že dynamická úloha bude řešena numerickými metodami pro řešení diferenciálních rovnic.

Nejprve shrňme řešený problém. Úkolem je řešit v určitém časovém intervalu

$$t = (t_0 \dots t_{kon}), \text{ kde } t_0 = 0 \quad (6.45)$$

dynamickou úlohu s počátečními podmínkami

$$\varphi_{t_0} = 0 \text{ a } \dot{\varphi}_{t_0} = 0. \quad (6.46)$$

Úloha je definována 5 pohybovými rovnicemi 6.32, 6.40 a 6.44 a 4 vztahy mezi zrychleními 6.10 a 6.17. Neznámými v těchto rovnicích jsou 4 reakce a 5 úhlových zrychlení

$$F_{Bmy}^{3R}, F_{Bmz}^{3R}, M_{3x}^{3R}, M_{3z}^{3R}, \ddot{\varphi}, \alpha_{2z,2}, \alpha_{13,3}. \quad (6.47)$$

Poloha mechanismu je dána maticovou rovnicí 6.4 a úhlové rychlosti jsou definovány vztahem 6.15. V následujícím textu bude řešení této dynamické úlohy navrženo dvěma metodami, kdy jedna z nich je navržena pro svoji jednoduchost a druhá metoda pro její vysokou přesnost. Podrobnější informace o algoritmech výpočtu pohybových rovnic lze nalézt v [38] a [39].

### 6.6.1 Modifikovaná metoda ARV

Metoda je založena na známé Eulerově metodě. Stejně jako Eulerova metoda i modifikovaná metoda ARV rozděluje řešení úlohy do časových intervalů

$$t = (t_0, t_0 + \Delta t, t_0 + 2 \cdot \Delta t \dots t_{kon}) \quad (6.48)$$

a v těchto nahrazuje nerovnoměrný pohyb hnacího členu (klika 4) pohybem rovnoměrně zrychleným v jednotlivých časových intervalech s tím, že v každém časovém intervalu je jiné zrychlení a rychlost hnacího členu.

Tato metoda je jednoduše programovatelná, avšak daň za tuto jednoduchost je nízká přesnost. Chyba metody je lineárně úměrná délce časového kroku, takže chybová funkce má tvar

$$err_{mod.ARV} = O(\Delta t^2). \quad (6.49)$$

Pro zvýšení přesnosti musíme zvyšovat počet kroků neboli snižovat délku časového kroku  $\Delta t$ . Srovnání této metody s přesnější metodou je uvedeno v následující kapitole.

#### Algoritmus VIII - Modifikovaná metoda ARV pro dynamickou úlohu

1. Nastavme: pořadové číslo iterace  $k = 0$ , počáteční odhad řešení úlohy polohy  $\mathbf{q}_0^{(k)}$ , časový krok  $\Delta t$ , čas  $t^{(k)} = t_0$ ,  $\varphi^{(k)} = \varphi_0$ ,  $\dot{\varphi}^{(k)} = \dot{\varphi}_{t_0}$ , parametry Martinezovy metody:  $maxit$ ,  $tol$ ,  $\varepsilon_{NS}$ .

2. Martinezovou hybridní metodou vypočteme úlohu polohy 6.4 s odhadem  $\mathbf{q}_0^{(k)}$

$$\mathbf{E} = \mathbf{T}_{12}^{(k)} \cdot \mathbf{T}_{23}^{(k)} \cdot \mathbf{T}_{34}^{(k)} \cdot \mathbf{T}_{41}^{(k)} \rightarrow \mathbf{q}^{(k)}. \quad (6.50)$$

3. Stanovení převodů a jejich derivací dle vztahů pro numerický výpočet derivace, viz kap. 3.3.10

$$\mathbf{p}_1^{(k)}, d\mathbf{p}_1^{(k)}, \mathbf{P}_{13}^{(k)}, \mathbf{DP}_{13}^{(k)}. \quad (6.51)$$

Pozor zde musí být dodrženo pravidlo z konce kap. 3.3.10 pro volbu kroku  $h$ , na kterém je určována numerická derivace

$$h \geq 10^6 \cdot \varepsilon_{NS}. \quad (6.52)$$

4. Stanovení úhlových zrychlení vidlice 3

$$\mathbf{V}_{13}^{(k)} = \left( \mathbf{T}_{13}^{(k)} \right)^{-1} \cdot \mathbf{P}_{13}^{(k)} \cdot \dot{\varphi}^{(k)} \quad (6.53)$$

5. Výpočet soustavy 12-ti lineárních rovnic. Soustava se sestává z 5 pohybových rovnic 6.32, 6.40 a 6.44 a 4 vztahů pro zrychlení 6.10 a 6.17. Získáme hodnoty neznámých

$$F_{Bmy}^{3R(k)}, F_{Bmz}^{3(k)}, M_{3x}^{3R(k)}, M_{3z}^{3R(k)}, \ddot{\varphi}^{(k)}, \alpha_{2z,2}^{(k)}, \alpha_{13,3}^{(k)}. \quad (6.54)$$

Pro výpočet této soustavy může být využitý jakýkoli řešič soustav lineárních rovnic nebo také navržený řešič soustav nelineárních rovnic, který ale bude méně efektivní.

6. Výpočet úhlu natočení hnacího členu 4 dle vztahu rovnoměrného pohybu

$$\varphi^{(k+1)} = \varphi^{(k)} + \dot{\varphi}^{(k)} \cdot \Delta t + \frac{1}{2} \ddot{\varphi}^{(k)} \cdot \Delta t^2. \quad (6.55)$$

7. Výpočet rychlosti na konci integračního kroku

$$\dot{\varphi}^{(k+1)} = \dot{\varphi}^{(k)} + \ddot{\varphi}^{(k)} \cdot \Delta t. \quad (6.56)$$

### 8. Příprava nové iterace

$$\begin{aligned} &\text{pokud } t^{(k)} > t_{kon} \text{ ukonči algoritmus} \\ &\text{jinak } \mathbf{q}_0^{(k+1)} = \mathbf{q}^{(k)}, \quad t^{(k+1)} = t^{(k)} + \Delta t, \quad k = k + 1. \end{aligned} \quad (6.57)$$

### 9. Pokračujme krokem 2.

## 6.6.2 Metoda Runge-Kutta 4. řádu

Metoda Runge-Kutta je jedna z nejpoužívanějších metod pro numerický výpočet diferenciálních rovnic. Zde bude prezentován algoritmus pro výpočet diferenciální rovnice druhého řádu. Bude uvažována varianta metody 4. řádu. Přesnost metody je úměrná páté mocnině časového kroku, takže chybová funkce je

$$err_{RK4} = O(\Delta t^5). \quad (6.58)$$

Nejprve zavedme subalgoritmus, který zjednoduší a zpřehlední algoritmus Runge-Kutta metody. Tato procedura stanoví hodnotu zrychlení kliky 4 pro zadanou polohu a rychlosti mechanismu.

Algoritmus IX - Subalgoritmus metody Runge-Kutta pro výpočet zrychlení

1. Vstupními veličinami metody jsou do metody jsou

$$t, \varphi, \dot{\varphi}, \mathbf{q}_0. \quad (6.59)$$

2. Martinezovou hybridní metodou vypočteme úlohu polohy 6.4 s odhadem  $\mathbf{q}_0^{(k)}$

$$\mathbf{E} = \mathbf{T}_{12} \cdot \mathbf{T}_{23} \cdot \mathbf{T}_{34} \cdot \mathbf{T}_{41} \rightarrow \mathbf{q}. \quad (6.60)$$

3. Stanovení převodů a jejich derivací dle vztahů pro numerický výpočet derivace, viz kap. 3.3.10

$$p_1, dp_1, \mathbf{P}_{13}, \mathbf{DP}_{13}. \quad (6.61)$$

Pozor zde musí být dodrženo pravidlo z konce kap. 3.3.10 pro volbu kroku  $h$ , na kterém je určována numerická derivace

$$h \geq 10^6 \cdot \varepsilon_{NS}. \quad (6.62)$$

4. Stanovení úhlových zrychlení vidlice 3

$$\mathbf{V}_{13} = (\mathbf{T}_{13})^{-1} \cdot \mathbf{P}_{13} \cdot \dot{\varphi} \quad (6.63)$$

5. Výpočet soustavy 12-ti lineárních rovnic. Soustava se sestává z 5 pohybových rovnic 6.32, 6.40 a 6.44 a 4 vztahů pro zrychlení 6.10 a 6.17. Získáme hodnoty neznámých

$$F_{Bmy}^{3R}, F_{Bmz}^3, M_{3x}^{3R}, M_{3z}^{3R}, \ddot{\varphi}, \alpha_{2z,2}, \alpha_{I3,3}. \quad (6.64)$$

Pro výpočet této soustavy může být využitý jakýkoli řešič soustav lineárních rovnic nebo také navržený řešič soustav nelineárních rovnic, který ale bude méně efektivní.

6. Výstupem subalgoritmu je hodnota úhlového zrychlení hnacího členu 4

$$\ddot{\varphi}. \quad (6.65)$$

Samotný algoritmus metody Runge-Kutta 4. řádu stanovuje v každém časovém intervalu  $\Delta t$  zrychlení ve 4 časových okamžicích a z těchto zrychlení poté stanovuje novou hodnotu úhlu natočení a rychlosti hnacího členu.

Algoritmus X - Algoritmus metody Runge-Kutta 4. řádu

1. Nastavme: pořadové číslo iterace  $k = 0$ , počáteční odhad řešení úlohy polohy  $\mathbf{q}_0^{(k)}$ , časový krok  $\Delta t$ , čas  $t^{(k)} = t_0$ ,  $\varphi^{(k)} = \varphi_0$ ,  $\dot{\varphi}^{(k)} = \dot{\varphi}_{t0}$ , parametry Martinezovy metody:  $maxit, tol, \varepsilon_{NS}$ .

2. Subalgoritmem IX se vstupními parametry

$$t = t^{(k)}, \varphi = \varphi^{(k)}, \dot{\varphi} = \dot{\varphi}^{(k)}, \mathbf{q}_0 = \mathbf{q}_0^{(k)}. \quad (6.66)$$

vypočteme zrychlení, které označíme

$$k_1 = \ddot{\varphi}. \quad (6.67)$$

Dále uložíme vypočtený vektor souřadnic vazeb získaný z úlohy polohy

$$\mathbf{q}^{(k)} = \mathbf{q}. \quad (6.68)$$

3. Subalgoritmem IX se vstupními parametry

$$t = t^{(k)} + \frac{\Delta t}{2}, \varphi = \varphi^{(k)} + \dot{\varphi}^{(k)} \cdot \frac{\Delta t}{2} + \frac{1}{2} k_1 \cdot \left( \Delta \frac{t}{2} \right)^2, \dot{\varphi} = \dot{\varphi}^{(k)} + k_1 \cdot \Delta \frac{t}{2}, \mathbf{q}_0 = \mathbf{q}_0^{(k)}. \quad (6.69)$$

vypočteme zrychlení, které označíme

$$k_2 = \ddot{\varphi}. \quad (6.70)$$

4. Subalgoritmem IX se vstupními parametry

$$t = t^{(k)} + \frac{\Delta t}{2}, \varphi = \varphi^{(k)} + \dot{\varphi}^{(k)} \cdot \frac{\Delta t}{2} + \frac{1}{2} k_2 \cdot \left( \Delta \frac{t}{2} \right)^2, \dot{\varphi} = \dot{\varphi}^{(k)} + k_2 \cdot \Delta \frac{t}{2}, \mathbf{q}_0 = \mathbf{q}_0^{(k)}. \quad (6.71)$$

vypočteme zrychlení, které označíme

$$k_3 = \ddot{\varphi}. \quad (6.72)$$



5. Subalgoritm IX se vstupními parametry

$$t = t^{(k)} + \Delta t, \varphi = \varphi^{(k)} + \dot{\varphi}^{(k)} \cdot \Delta t + \frac{1}{2} k_3 \cdot (\Delta t)^2, \dot{\varphi} = \dot{\varphi}^{(k)} + k_3 \cdot \Delta t, \mathbf{q}_0 = \mathbf{q}_0^{(k)}. \quad (6.73)$$

vypočteme zrychlení, které označíme

$$k_4 = \ddot{\varphi}. \quad (6.74)$$

6. Vypočteme úhel natočení a počáteční rychlost pro další iteraci

$$\begin{aligned} \varphi^{(k+1)} &= \varphi^{(k)} + \dot{\varphi}^{(k)} \cdot \Delta t + (k_1 + k_2 + k_3) \cdot \frac{\Delta t^2}{6}, \\ \dot{\varphi}^{(k+1)} &= \dot{\varphi}^{(k)} + (k_1 + 2 \cdot k_2 + 2 \cdot k_3 + k_4) \cdot \frac{\Delta t}{6}. \end{aligned} \quad (6.75)$$

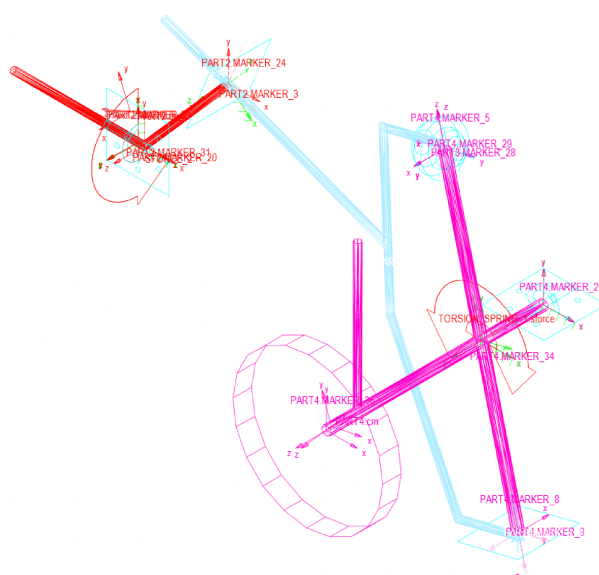
7. Příprava nové iterace

$$\begin{aligned} &\text{pokud } t^{(k)} > t_{kon} \text{ ukonči algoritmus} \\ &\text{jinak } \mathbf{q}_0^{(k+1)} = \mathbf{q}^{(k)}, \quad t^{(k+1)} = t^{(k)} + \Delta t, \quad k = k + 1. \end{aligned} \quad (6.76)$$

8. Pokračujeme krokem 2.

## 6.7 Srovnání výsledků dynamické úlohy

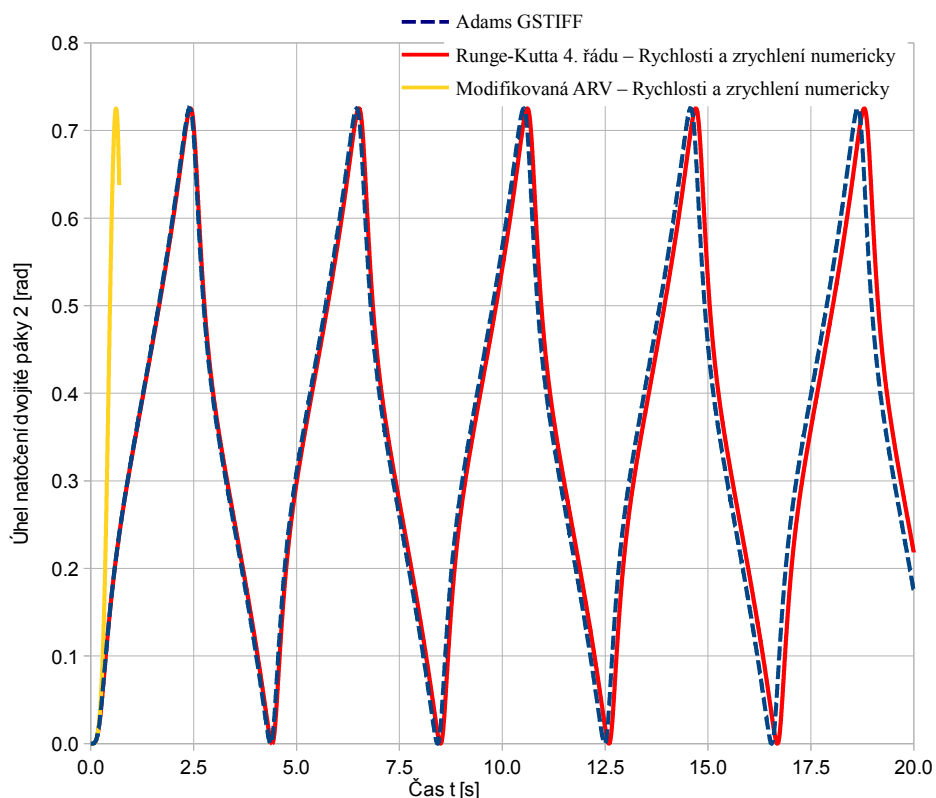
Dynamika prohozního mechanismu, jak je zobrazen na obr. 6.2, byla simulována také v multibody softwaru ADAMS 2011. Algoritmus řešící numericky diferenciální pohybovou rovnici byl GSTIFF. Jedná se o výrobcem přednastavenou metodu. U této metody nenalezl autor vyjádření její chybové funkce. Model mechanismu v programovém prostředí je zobrazen na obr. 6.6.



Obrázek 6.6: Simulace prohozního mechanismu v multibody softwaru ADAMS 2011

Výsledky získané pomocí navrženého postupu, který využívá k výpočtu rychlostí a zrychlení numericky stanovené převodové funkce, byly srovnávány s výsledky získanými z multibody prostředí. Časový krok byl u všech metod

$$\Delta t = 10^{-3}. \quad (6.77)$$



Obrázek 6.7: Srovnání výsledného úhlu natočení v závislosti na čase

Srovnání je uvedeno pro kinematické veličiny dvojité páky 2. Na obr. 6.7 je uvedeno srovnání výsledného úhlu natočení v průběhu 20 s. Z grafu zřejmě vyplývá, že výsledky získané modifikovanou metodou ARV se značně liší od ostatních dvou metod. Dále se tato metoda ukázala jako silně nestabilní, protože již v čase 0.7 se začaly objevovat numerické potíže. Matice koeficientů lineární soustavy rovnic definovaná v 6.54 se stávala blízká singulární matici, což bylo identifikováno číslem podmíněnosti  $\kappa$  dle kap. 5.7. Tento efekt nebyl odstraněn ani zkracováním kroku. Nejmenší testovaný krok měl délku

$$\Delta t = 10^{-5}. \quad (6.78)$$

Z těchto důvodů nelze modifikovanou metodu ARV doporučit. Její použití by bylo přípustné pouze při dynamických dějích, kdy změny zrychlení jsou malé. Tento závěr byl odvozen z toho, že na počátku, kdy zrychlení zvolna stoupá, tak jsou všechny křivky úhlu natočení páky 2 shodné.

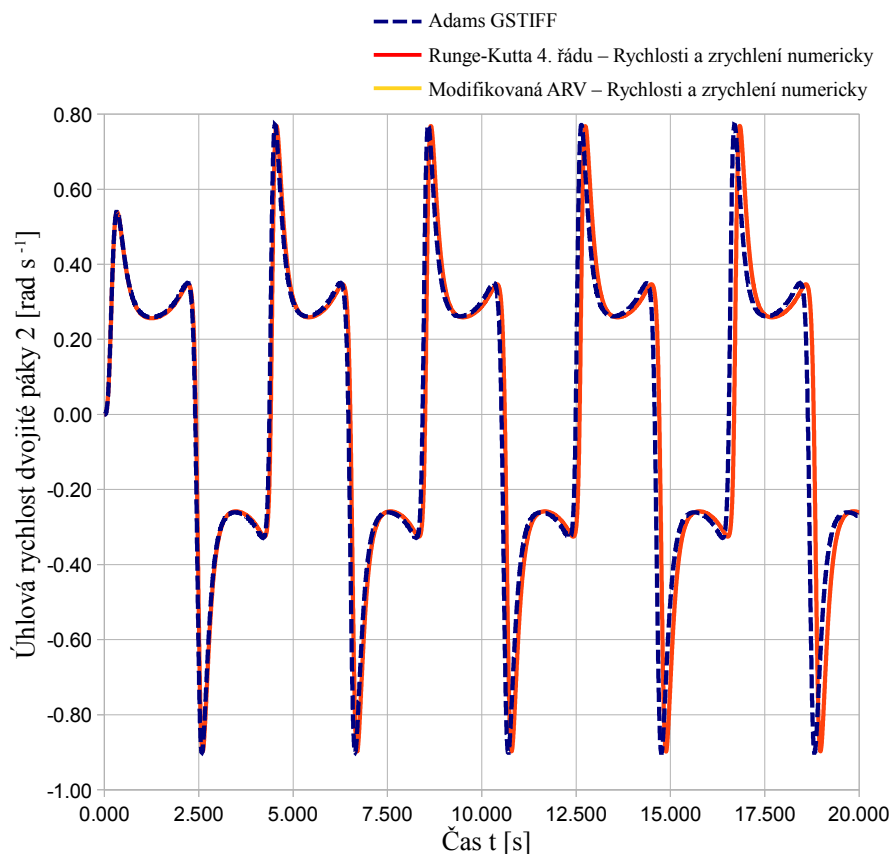
Pokud srovnáme výsledky získané ze softwaru ADAMS a výsledky získané navrhovaným postupem při použití Runge-Kutta 4. řádu, viz algoritmus XI, můžeme prohlásit, že

tvary křivky je zcela shodný. Toto platí pro úhel natočení, úhlovou rychlost i úhlové zrychlení dvojité páky 2, viz obr. 6.7, 6.8 a 6.9. Tyto dvě křivky se ve všech případech vzájemně časově posouvají. Časový posun a konci děje je

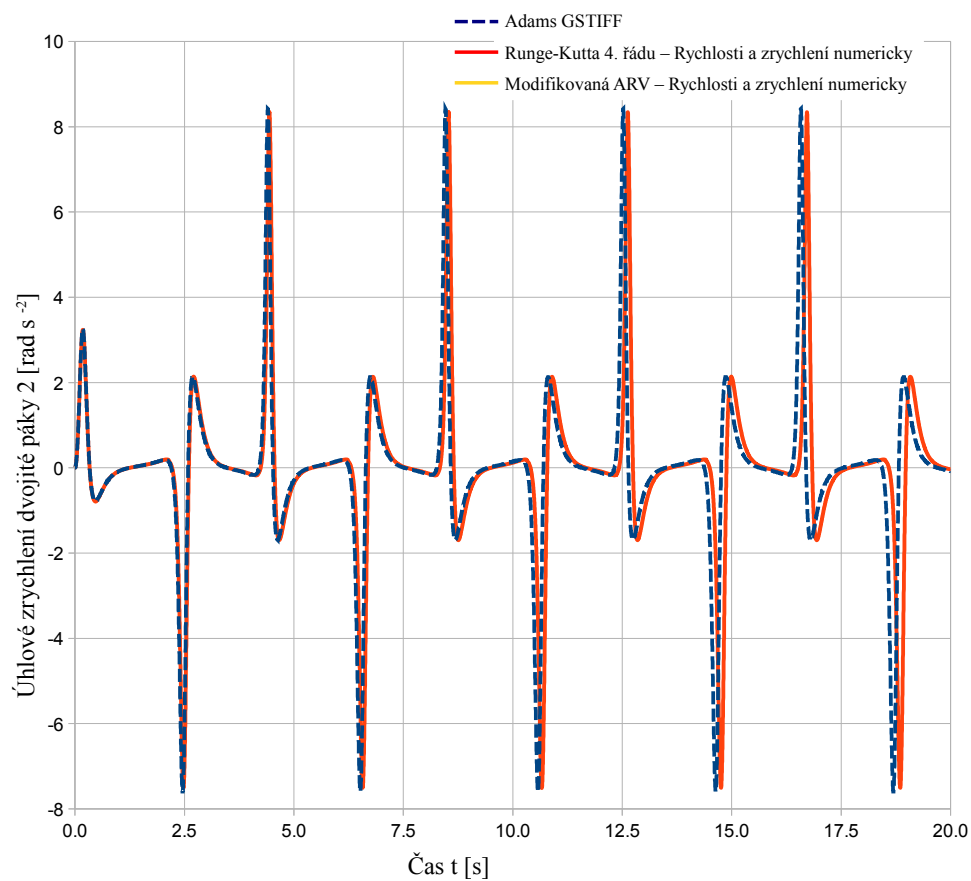
$$\text{posun} = 0.141 \text{ s.} \quad (6.79)$$

Lze obtížně identifikovat, který model je přesnější a čím je časový posun výsledků způsoben. Prvním důvodem vzniku tohoto posunu může být použití rozdílných metod numerického řešení diferenciální rovnice. Dalším původcem chyby může být zaokrouhlování vstupních parametrů. Vliv užití numerické derivace lze zmírnit použitím formulí vyššího řádu dle tab. 3.3.

Hodnotu časového posunu však pokládáme za dostatečně malou a dále s přihlédnutím ke shodným tvarům všech křivek hodnotíme navržený způsob výpočtu dynamické úlohy jako vhodný. Užití převodových funkcí transformačních matic vypočtených numerickou derivací značně zjednodušuje práci řešiteli a je otázkou, zda chyba vnesená vztahem 3.51 a 3.52, který vyžaduje vysoké množství násobení matic, není větší. Také je zapotřebí vzít v úvahu, že při použití rovnic 3.51 a 3.52 by stoupl počet neznámých a rovnic v 6.64, protože by nebylo možno stanovit pouze vybrané rychlosti a zrychlení, jak je možné v případě numericky stanovených převodových funkcí. Vzhledem k tomu, že toto není hlavním cílem práce, nebyly výše uvedené vlivy dále zkoumány.



Obrázek 6.8: Srovnání úhlové rychlosti páky 2 v závislosti na čase



Obrázek 6.9: Srovnání úhlového zrychlení páky 2 v závislosti na čase

## 7 Levostranný mechanismus řazení

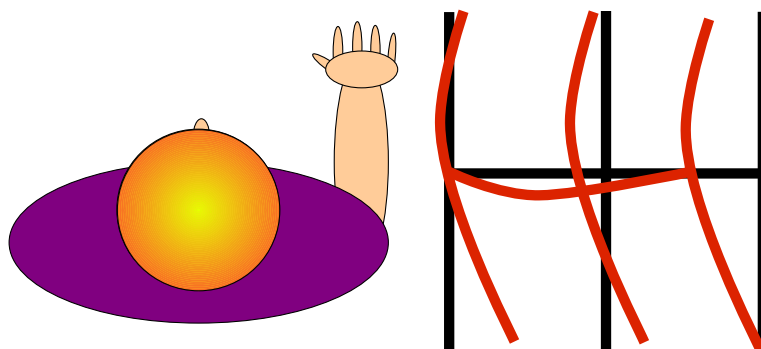
V dnešní době, kdy nabídka nákladních vozů určených do extrémních podmínek je velmi pestrá a široká, se spotřebitel rozhoduje při výběru vozu podle mnoha kritérií a jedním z nich, zdaleka ne bezvýznamným, je hledisko ergonomické. Pro zákazníka je podstatné, aby ovládání a manipulace s automobilem byla v co největším souladu s lidskými pohybovými možnostmi. Jestliže se podaří dosáhnout takového souladu, zdá se ovládání automobilu řidiči přirozené a intuitivní, a tudíž pohodlné a méně únavné.

Jedním z mnoha konstrukčních prvků nákladního automobilu je mechanismus řazení, který slouží řidiči k volbě rychlostních stupňů. Význam ergonomického hlediska je v tomto případě neoddiskutovatelný.

Přenos pohybu řidičovy ruky na převodovou skříň může být založen na několika principech. Asi nejmodernější strategií je využití elektroniky, kdy řadicí páka je nahrazena joystickem. Elektronický mechanismus však s sebou nese riziko poruchovosti, a to zvláště v případě použití vozu v extrémních podmínkách; tento je tudíž princip nepoužitelný.

Druhou možností je použití ocelového lanka. Bohužel i tato konstrukční varianta má hned dva podstatné nedostatky. Při tlakovém namáhání lanka, ke kterému u lankových mechanismů řazení dochází, není přenos pohybu přesný. Druhým nedostatkem je, že během provozu se mezi lanko a jeho vedení dostávají nečistoty jako bláto a písek, které mohou zapříčinit v krajních případech až ztrátu funkčnosti mechanismu.

Zbývá tudíž přenést pohyb pomocí tuhých členů. Nevýhodou tohoto typu mechanismů je jejich značná strukturální složitost, která bohužel způsobuje nepřirozené zakřivení cesty řadicí páky. Příklad takového zakřivení je uveden na obr. 7.1 červenou křivkou.



Obrázek 7.1: Příklad ideálního (černě) a skutečného (červeně) tvaru drah řazení

Cílem této části práce bude tudíž optimalizace dvou typů mechanismů řazení s tuhými

členy tak, aby tvar trajektorie rukojeti řadicí páky byl přirozený. Ideální dráhy řazení, které jsou na obrázku znázorněny černou čarou, mají přímý a nezakřivený tvar. Důležitou částí práce bude také analýza stability pohybu mechanismu vzhledem ke vzniklým výrobním a montážním imperfekcím.

## 7.1 Rozdělení mechanismů řazení na levostranné a pravostranné

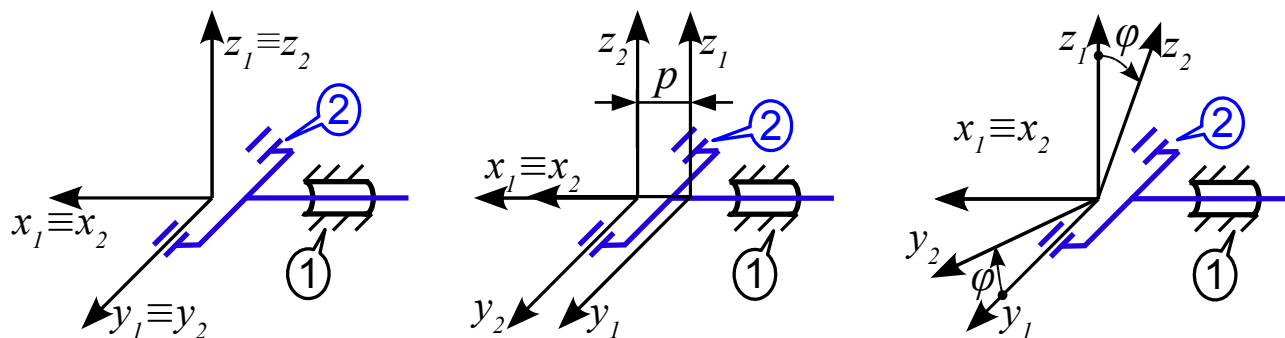
Mechanismus řazení nákladního vozu slouží řidiči k volbě rychlostních stupňů převodové skříně. Mechanismy lze rozdělit podle toho, zda jsou určeny pro vozidla s řidičem sedícím vlevo nebo vpravo na:

1. Levostranné mechanismy, kdy řidič sedí vlevo a má řadicí páku po pravé ruce, používá se u vozů určených k provozu v zemích, kde se jezdí po pravé straně vozovky, např. v České republice.
2. Pravostranné mechanismy, kdy řidič sedí vpravo a má řadicí páku po levé ruce. Tyto vozy jsou určeny pro země, kde se jezdí po levé straně vozovky, např. Velkou Británii nebo Indii.

## 7.2 Souřadnice mechanismů řazení

Pro pohyby řadicí pákou, kterými řidič volí rychlostní stupně, budou použity názvy *řazení*, což je pohyb pákou ve směru jízdy automobilu, pohyb kolmý na směr jízdy bude označován jako *volba*.

Převodová skříně je ovládána pomocí dvou vstupních souřadnic respektive dvou souřadnic mechanismu, posuvu  $p$  a natočení  $\varphi$  kolem téže osy, po které se koná posuv. Každému rychlostnímu stupni odpovídá jistá kombinace hodnot těchto dvou souřadnic, viz. kap. 7.3 a 8.2. Protože mechanismy řazení slouží právě k ovládání převodové skříně, jsou tyto dva stupně volnosti tzv. souřadnicemi mechanismu řazení, což znamená, že tyto dvě souřadnice jednoznačně určují polohu celého mechanismu.



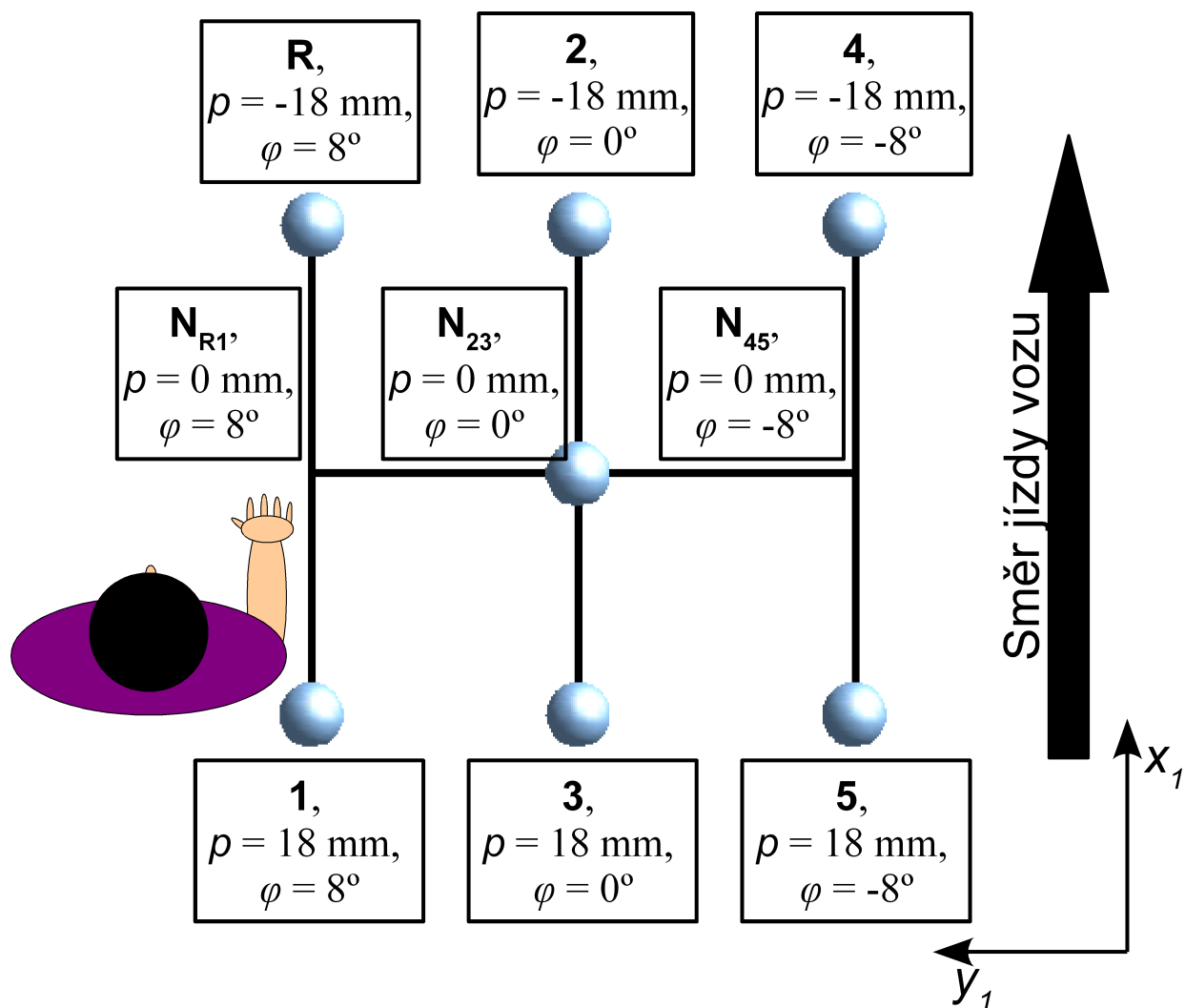
Obrázek 7.2: Přenos souřadnic mechanismu na vidlici 2

Souřadnice mechanismu jsou u obou typů mechanismu aplikovány na vidlici 2, viz obr. 7.2. Souřadný systém 1 přísluší rámu, čili pevnému prostoru, a naopak souřadný systém 2 vidlici 2, která je s rámem 1 spojena válcovou vazbou.

Pozn.: Orientace souřadných systémů v obrázku odpovídá levostrannému mechanismu, u pravostranného byla použit jiný souřadný systém (posuv a rotace se koná kolem osy  $y_1$ ), princip přenosu souřadnic mechanismu na vidlici 2 však zůstává stejný.

Dále lze říci, že změna souřadnice  $p$  způsobuje pohyb řadicí páky zvaný řazení a změna souřadnice  $\varphi$  pohyb zvaný volba.

### 7.3 Rozložení rychlostních stupňů převodové skříně ovládané levostranným mechanismem a hodnoty souřadnic mechanismu



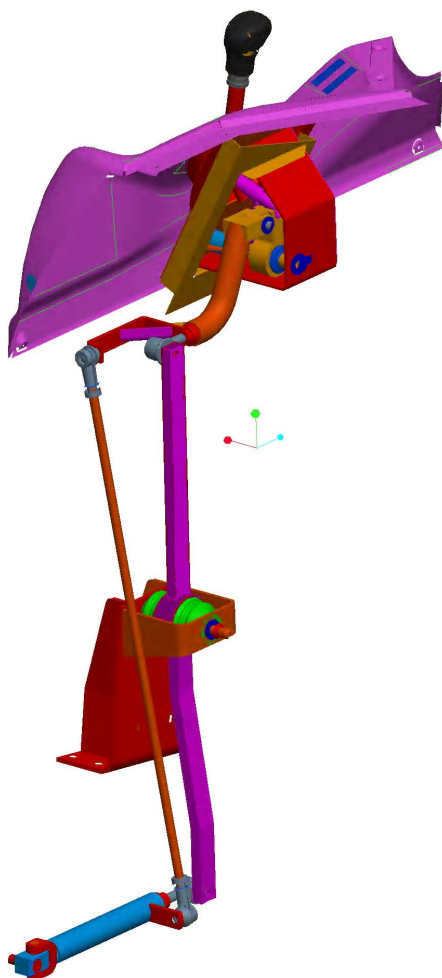
Obrázek 7.3: Rozložení rychlostních stupňů a bodů volby na řadicí páce 8 a jim odpovídající hodnoty souřadnic mechanismu

Levostranný mechanismus je optimalizován pro ovládání pětistupňové převodové

skříně, která je ovládána pomocí dvou vstupních souřadnic (též souřadnice mechanismu)  $p$  a  $\varphi$ . Polohy rukojeti řadicí páky v pevném souřadném systému 1, které přísluší rychlostním stupňům s odpovídajícími hodnotami souřadnic mechanismu, jsou uvedeny na obr. 7.3. Písmenem  $R$  je označen zpětný chod, písmenem  $N$  jsou označeny významné body pohybu zvaného volba a čísla označují rychlostní stupně. Neutrál vozidla je umístěn v bodě  $N_{23}$ . Na obrázku je též schematicky naznačena poloha řidiče a jeho řadicí pravé ruky.

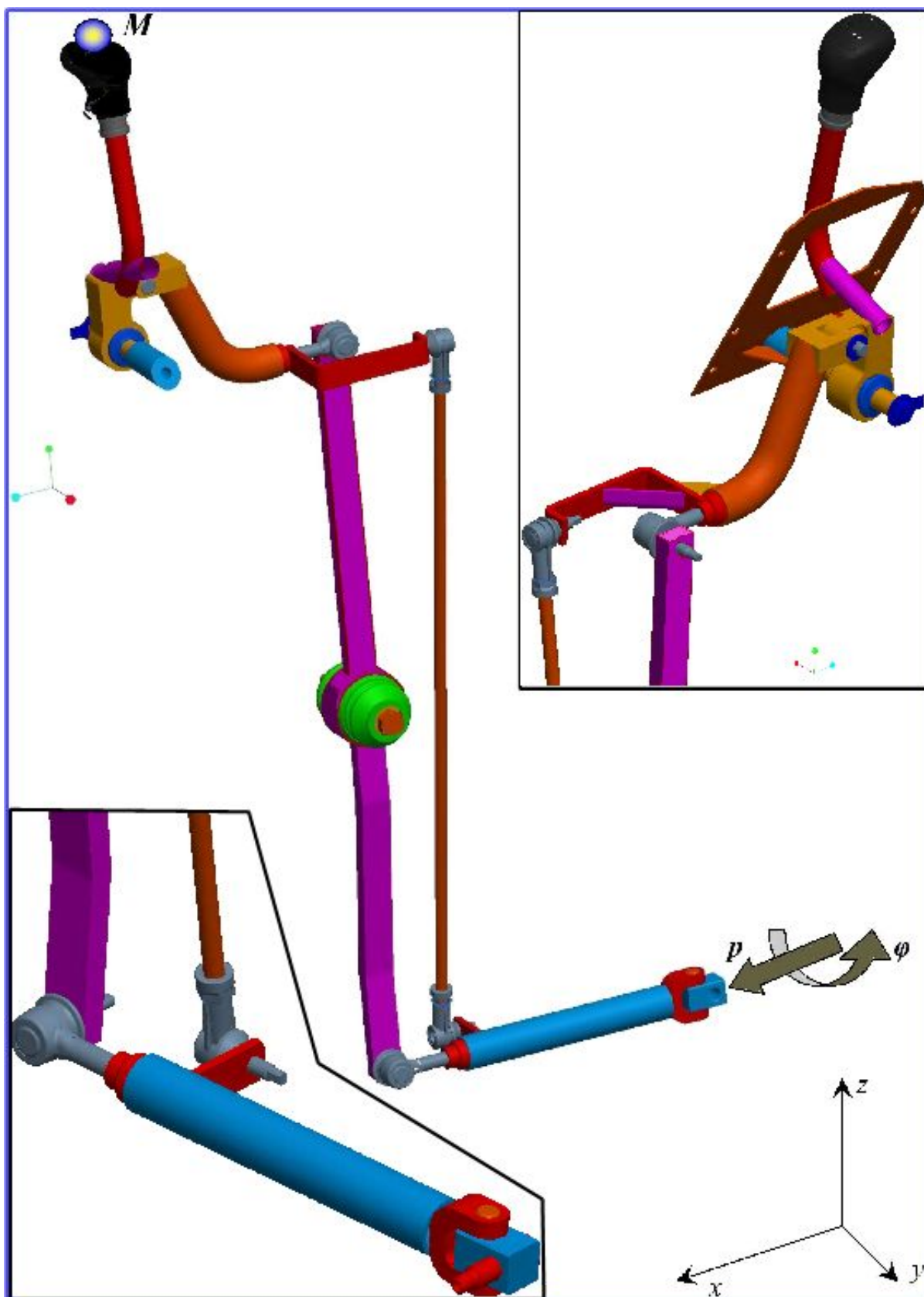
## 7.4 Kinematický rozbor funkce levostranného mechanismu řazení

Prostorový model levostranného mechanismu řazení vytvořený v programu Pro/ENGINEER je vyobrazen na následujících obr. 7.4 a 7.5 včetně detailů důležitých částí. Pro lepší představu o pohybu levostranného mechanismu řazení byla vytvořena videosekvence, která je přiložena na CD. Bod  $M$  bude vždy značit bod na rukojeti řadicí páky, tzn. bod, jehož trajektorie bude optimalizována.



Obrázek 7.4: Prostorový model včetně součástí, které mechanismus spojují s rámem

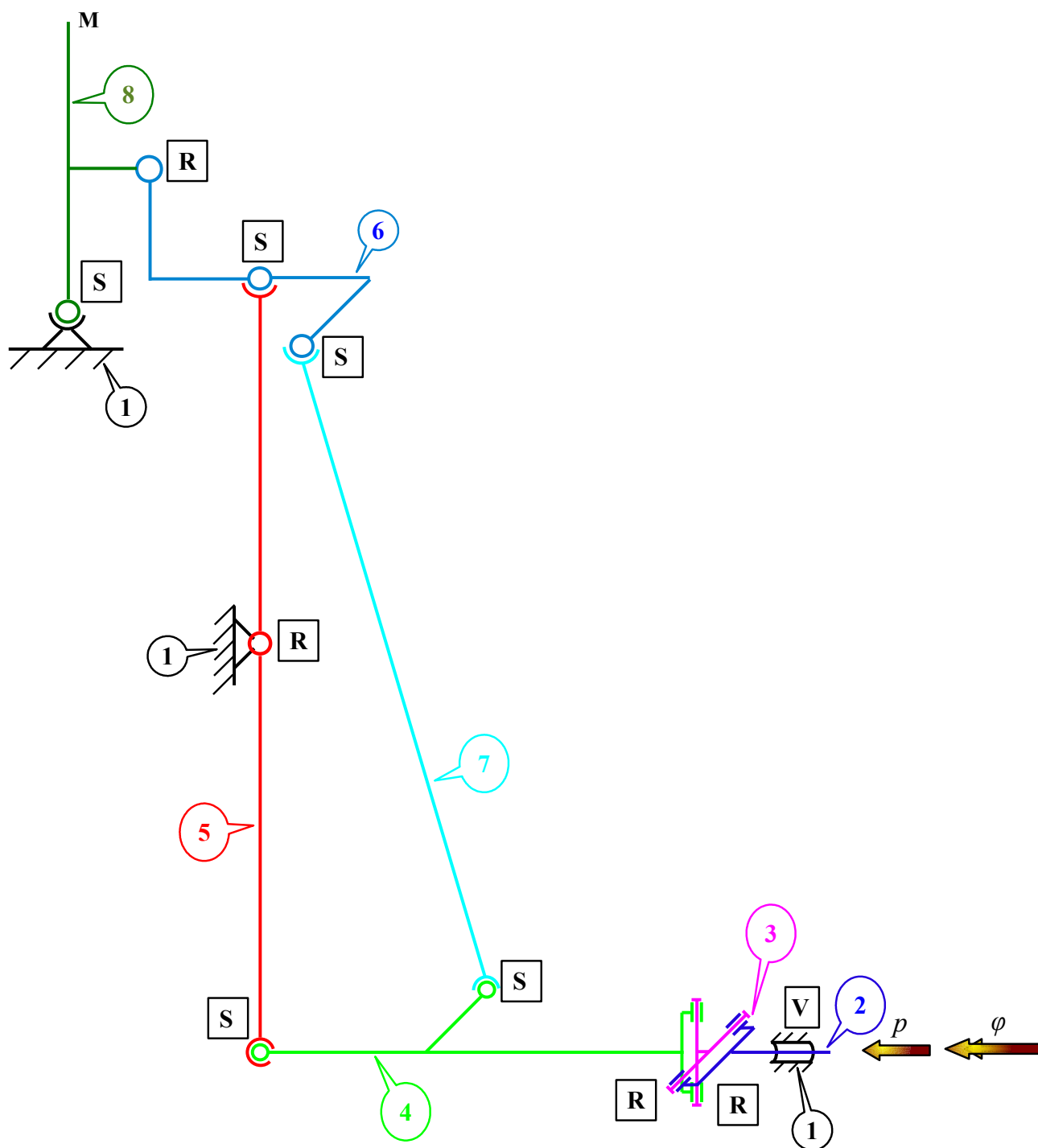




Obrázek 7.5: Prostorový model levostranného mechanismu řazení s detaily vybraných částí

Pro lepší přehled o funkci mechanismu bylo vytvořeno kinematické schéma, viz obr. 7.6. Každé těleso je označeno pořadovým číslem, které je vyznačeno v bublině u tělesa. Druhy

vazeb jsou označovány písmeny v obdélníku. Písmeno V značí válcovou vazbu umožňující posuv po ose a rotaci kolem téže osy, písmeno R vazbu rotační a písmeno S sférickou kinematickou dvojici.



Obrázek 7.6: Kinematické schéma levostranného mechanismu řazení

Jak již bylo zmíněno výše, na vidlici 2, která je s rámem 1 spojena válcovou vazbou, jsou aplikovány souřadnice mechanismu, viz obr. 7.2. Dále je vidlice 2 vázána k tělesu 4 přes křížový člen 3, který není rovinný; toto spojení těles 2 a 4 je často nazýváno Hookovým kloubem.

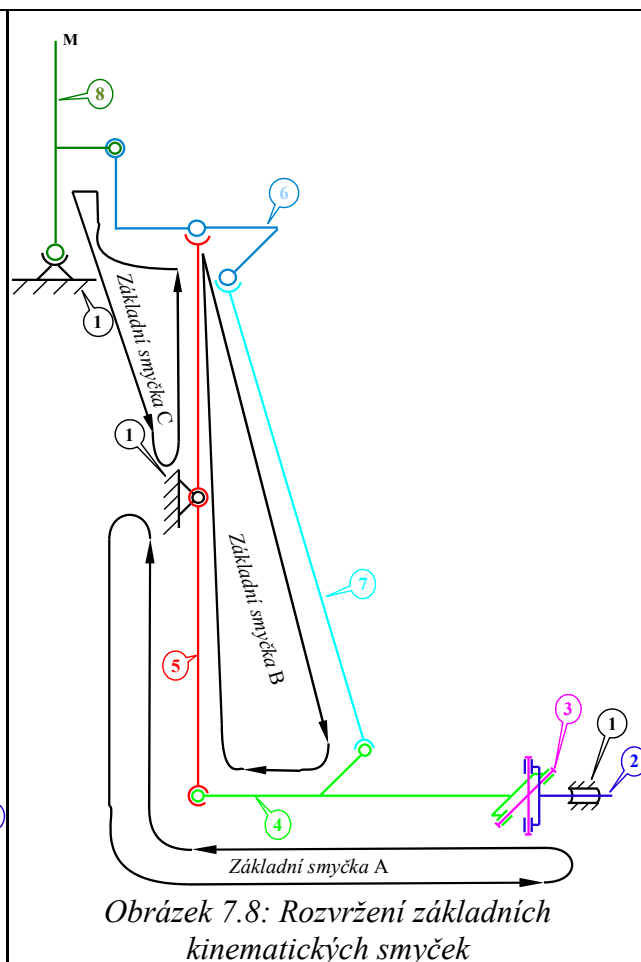
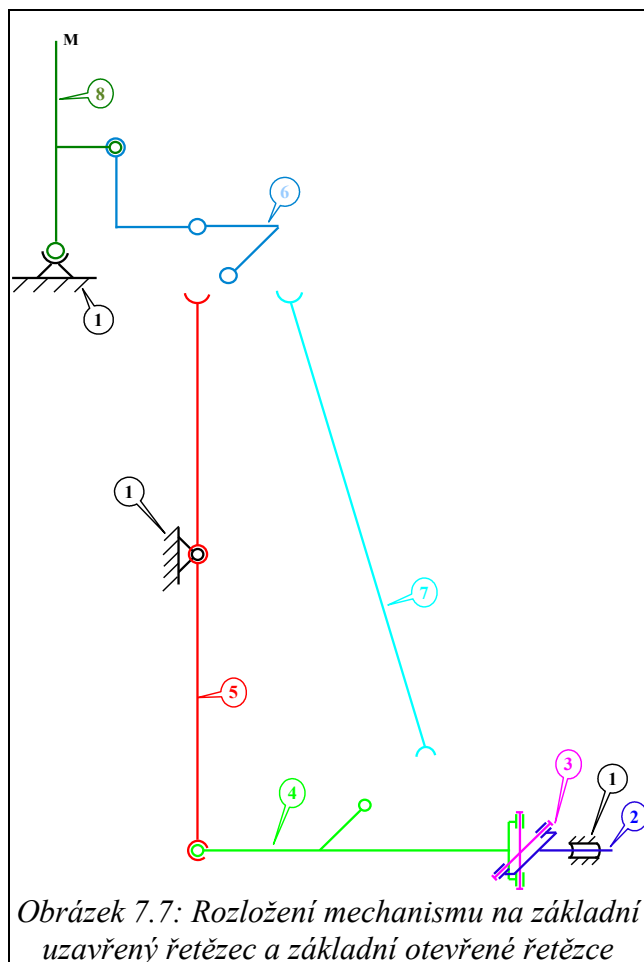
Pohyb řazení je přenášen na člen 6 z členu 4 pomocí páky 5, která je rotačně spojena s rámem 1. Pohyb volby je přenášen na člen 6 z členu 4 pomocí tyče 7. Oba pohyby, volba i řazení, jsou z členu 6 přenášeny na řadicí páku 8 rotační vazbou. Řadicí páka 8 je na rámu 1 uložena sférickým kloubem. Bohužel nejsou pohyby volby a řazení na sobě nezávislé, což způsobuje naklopení a zakřivení trajektorií budu  $M$ ; tento jev bude podrobněji popsán dále.

## 7.5 Stanovení počtu stupňů volnosti a počtu základních kinematických smyček

Počet stupňů volnosti  $i$ , byl určen vztahu 3.2

$$i = 6 \cdot (n - 1) - 3 \cdot S - 4 \cdot V - 5 \cdot R = 6 \cdot (8 - 1) - 3 \cdot 5 - 4 \cdot 1 - 5 \cdot 4 = 3, \quad (7.1)$$

kde velká písmena označují počet vazeb daného typu dle kinematického schématu z obr. 7.6 a písmeno  $n$  označuje počet těles včetně rámu. Z vypočtených třech stupňů volnosti dva připadají souřadnicím mechanismu a zbývající jeden stupeň volnosti odpovídá nedefinovanému otáčení tyče 7 kolem své podélné osy.



Pro levostranný mechanismus byl definován dle kap. 3.2.4 základní uzavřený řetězec

tvořený členy 1, 2, 3, 4 a 5 a další dva otevřené základní řetězce, viz obr. 7.7 a 7.8. První otevřený základní řetězec je složen z členů 6 a 8 a druhý základní otevřený řetězec je tvořen členem 7, celkový počet základních řetězců je tudíž roven třem, což je současně i počet základních smyček mechanismu. Rozložení základních smyček bylo zvoleno následující:

- A. Základní smyčka A je složena ze členů 1, 2, 3, 4 a 5.
- B. Základní smyčka B je složena ze členů 4, 5, 6 a 7.
- C. Základní smyčka C je složena ze členů 1, 5, 6 a 8.

## 7.6 Sestavení maticových rovnic úlohy polohy levostranného mechanismu řazení

Počet neznámých souřadnic vazeb, které budou určovány úlohou polohy, je

$$r \cdot 6 = 18, \quad (7.2)$$

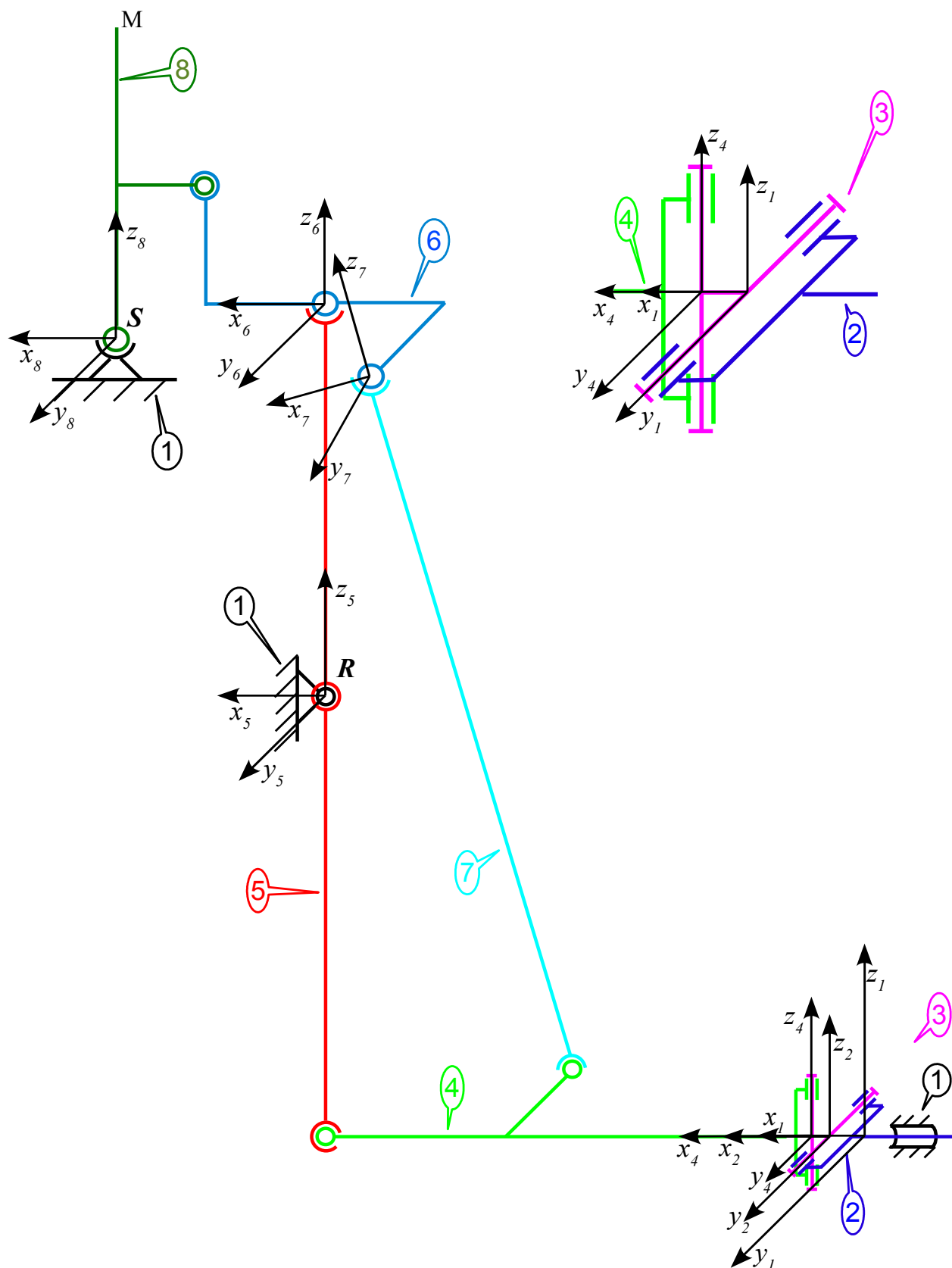
kde

$$r = 3 \quad (7.3)$$

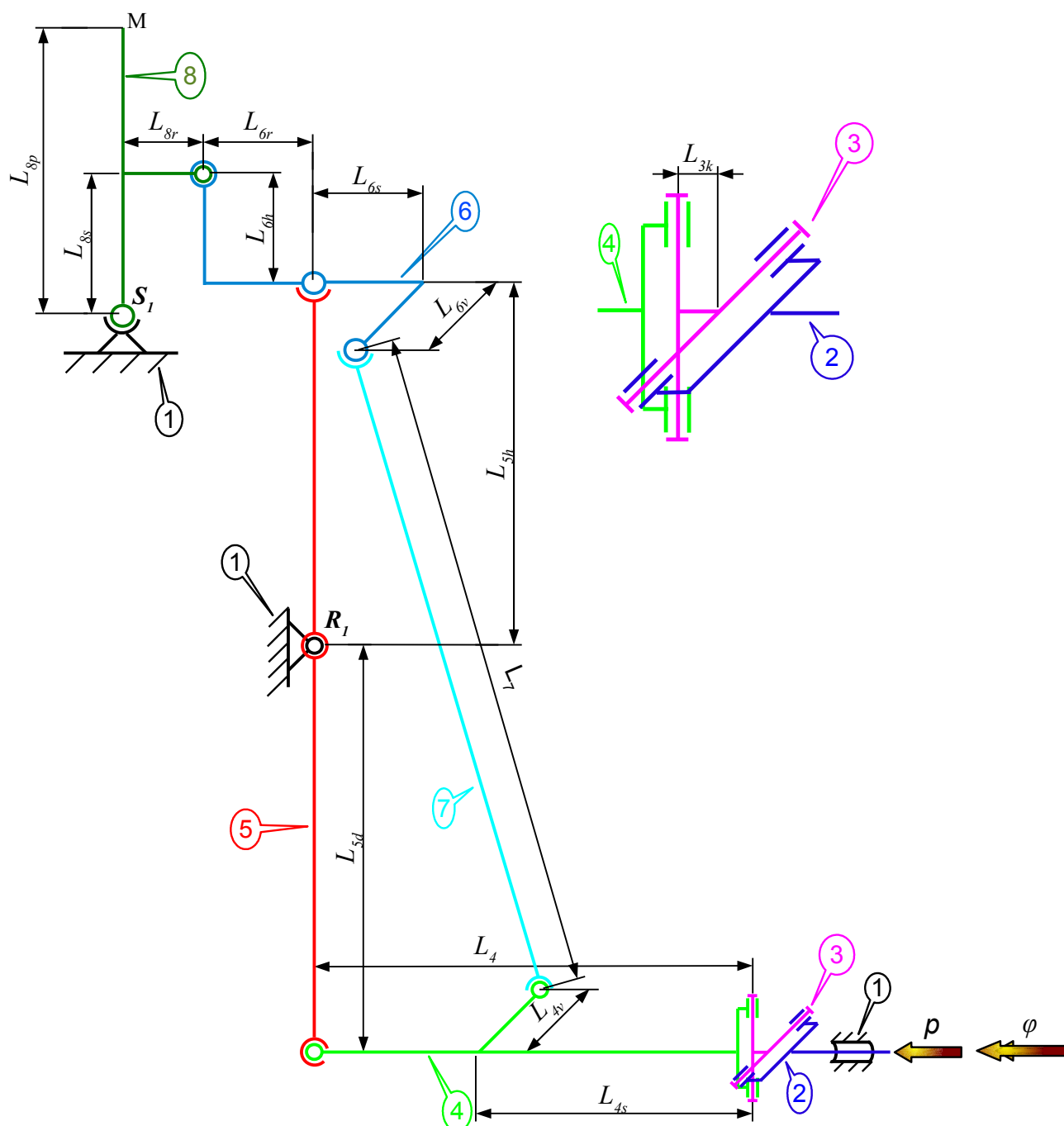
je počet základních smyček.

Prvním krokem při stanovování úlohy polohy maticovou metodou je zavedení tělesových souřadných systémů, což je provedeno na schématu z obr. 7.9. Pevný, tzn. nepohyblivý neboli globální souřadný systém 1 přísluší rámu 1 a je situován tak, že osa  $x_1$  je shodná s osou, po které respektive kolem které se děje posuv  $p$  respektive natočení  $\varphi$ . V počáteční poloze mechanismu (tzn.  $p = 0$  a  $\varphi = 0$ ) je osa  $y_1$  shodná s osou rotační vazby spojující vidlici 2 a křížový člen 3. Značení souřadných systémů dalších těles je analogické, pořadové číslo tělesa odpovídá pořadovému číslu tělesového souřadného systému jemu příslušejícímu. V pravé horní části schématu je vyobrazen detail Hookova kloubu.

Dalším krokem při sestavování úlohy polohy je popis rozměrů mechanismu, označení všech potřebných rozměrů je uvedeno na obr. 7.10. Nezbytným údajem jsou také souřadnice bodů, ve kterých je mechanismus vázán s rámem (což jsou středy sférických vazeb  $\mathbf{R}_1$  a  $\mathbf{S}_1$ ), v pevném souřadném systému 1. Dále je třeba upozornit, že táhlo o délce  $L_{4v}$  je kolmé k ose vidlice členu 4 a jeho délce a táhlo o délce  $L_{6v}$  je kolmé ke zbylé části členu 6. Detail Hookova kloubu je uveden na obrázku vpravo nahoře; z tohoto detailu je zřejmé, že křížový člen není rovinný, osy rotačních vazeb jsou mimoběžné a jejich nejkratší vzdálenost je  $L_{3k}$ . Souřadnice mechanismu posuv  $p$  a natočení  $\varphi$  jsou také vyznačeny.



Obrázek 7.9: Zavedení pevného a tělesových souřadných systémů sloužících ke stanovení úlohy polohy



Obrázek 7.10: Definice rozměrů levostranného mechanismu řazení

Maticové rovnice budou sestavovány pro každou základní smyčku zvlášť, jak bylo popsáno v kap. 3.3.8. Rozvržení těchto základních smyček bylo uvedeno na obrázku 7.8, zavedené souřadné systémy na obrázku 7.9 a rozměry byly popsány na obrázku 7.10.

### 7.6.1 Maticová rovnice základní smyčky A

Maticová rovnice základní smyčky A bude mít dle rovnice 3.44 následující tvar

$$T_{14} \cdot T_{45} \cdot T_{51} = E, \quad (7.4)$$

kde  $E \in \mathbb{R}^{4 \times 4}$  je matice identity a  $T_{ij}$  jsou transformační matice vyjadřující přechod z  $i$  – tého do  $j$ –tého tělesového souřadného systému; tyto matice lze vyjádřit pomocí součinu transformačních matic základních pohybů, dle kap. 3.3.2 ,

$$\begin{aligned} T_{14} &= T_x(p) \cdot R_x(\varphi) \cdot R_y(q_1) \cdot T_x(L_{3k}) \cdot R_z(q_2), \\ T_{45} &= T_x(L_4) \cdot R_x(q_3) \cdot R_y(q_4) \cdot R_z(q_5) \cdot T_z(L_{5d}), \\ T_{51} &= R_y(q_6) \cdot T_x(-x_{R1}) \cdot T_y(-y_{R1}) \cdot T_z(-z_{R1}). \end{aligned} \quad (7.5)$$

Při vyjadřování transformačních matic je zaveden vektor souřadnic vazeb  $q \in \mathbb{R}^{18}$ , jehož prvky jsou neznámé úhly natočení v jednotlivých vazbách, takže například  $q_1$  je úhel natočení křížového členu 3 vůči členu 2.

### 7.6.2 Maticová rovnice základní smyčky B

Maticová rovnice základní smyčky B bude mít dle rovnice 3.44 následující tvar

$$T_{45} \cdot T_{56} \cdot T_{67} \cdot T_{74} = E, \quad (7.6)$$

matice  $T_{45}$  byla již uvedena výše, viz. 7.4, a ostatní transformační matice lze opět rozepsat jako součin matic základních pohybů

$$\begin{aligned} T_{56} &= T_z(L_{5h}) \cdot R_x(q_7) \cdot R_y(q_8) \cdot R_z(q_9), \\ T_{67} &= T_x(-L_{6s}) \cdot T_y(L_{6v}) \cdot R_x(q_{10}) \cdot R_y(q_{11}) \cdot R_z(q_{12}), \\ T_{74} &= T_z(-L_7) \cdot R_x(q_{13}) \cdot R_y(q_{14}) \cdot T_y(L_{4v}) \cdot T_x(-L_{4s}). \end{aligned} \quad (7.7)$$

Ve vztahu pro matici  $T_{47}$  je vynechána rotace kolem osy  $z$  (tělesa 4 a 7 jsou vázána sférickou vazbou, která má tři stupně volnosti, jímž odpovídají tři natočení), čímž se eliminuje volné otáčení tyče 7 kolem její podélné osy.

### 7.6.3 Maticová rovnice základní smyčky C

Maticová rovnice základní smyčky C bude mít dle rovnice 3.44 následující tvar

$$T_{15} \cdot T_{56} \cdot T_{68} \cdot T_{81} = E. \quad (7.8)$$

Tvar matice  $T_{56}$  je uveden výše v 7.7, ostatní transformační matice potom budou mít tvar

$$\begin{aligned} T_{15} &= T_x(x_{R1}) \cdot T_y(y_{R1}) \cdot T_z(z_{R1}) \cdot R_y(-q_6), \\ T_{68} &= T_x(L_{6r}) \cdot T_z(L_{6h}) \cdot R_y(q_{15}) \cdot T_x(L_{8r}) \cdot T_z(-L_{8s}), \\ T_{81} &= R_x(q_{16}) \cdot R_y(q_{17}) \cdot R_z(q_{18}) \cdot T_x(-x_{S1}) \cdot T_y(-y_{S1}) \cdot T_z(-z_{S1}). \end{aligned} \quad (7.9)$$

### 7.6.4 Shrnutí úlohy polohy

Sestavením třech maticových rovnic 7.4, 7.5 a 7.6 pro základní smyčky A, B a C jsme získali soustavu osmnácti transcendentních rovnic, kde jako neznámé vystupuje osmnáct prvků vektoru  $\mathbf{q} \in \mathbb{R}^{18}$  představující úhly natočení v jednotlivých vazbách. Tato soustava nelineárních rovnic byla řešena pomocí Martinezovy hybridní metody dle kap. 5.3.

### 7.6.5 Postup výpočtu trajektorie bodu M

Jakmile je vypočtena úloha polohy, lze stanovit polohu bodu  $\mathbf{M}$  rukojeti řadicí páky 8 v pevném souřadném systému 1 pomocí rozšířeného průvodiče bodu  $\mathbf{M}$  jako

$$\mathbf{r}_{1M} = \mathbf{T}_{18} \cdot \mathbf{r}_{8M}, \quad (7.10)$$

kde  $\mathbf{T}_{18}$  lze rozepsat jako

$$\mathbf{T}_{81} = \mathbf{T}_z(z_{S1}) \cdot \mathbf{T}_y(y_{S1}) \cdot \mathbf{T}_x(x_{S1}) \cdot \mathbf{R}_z(-\alpha_{18}) \cdot \mathbf{R}_y(-\alpha_{17}) \cdot \mathbf{R}_x(-\alpha_{16}), \quad (7.11)$$

a prvky rozšířeného průvodiče  $\mathbf{r}_{8M}$  bodu  $\mathbf{M}$  v tělesovém souřadném systému 8 řadicí páky jsou známy, struktura vektoru je následující

$$\mathbf{r}_{8M} = \begin{bmatrix} 0 \\ 0 \\ L_{8p} \\ 1 \end{bmatrix}. \quad (7.12)$$

Postup výpočtu trajektorie bodu  $\mathbf{M}$  při řazení rychlostních stupňů spočívá ve výpočtu úlohy polohy (dle rovnic získaných ze základních smyček) pro ty souřadnice mechanismu, které odpovídají rychlostním stupňům a bodům volby, viz obr. 7.3, a následném výpočtu polohy bodu  $\mathbf{M}$  dle vztahu 7.10. Tyto body postačují ke stanovení hodnoty cílové funkce; pokud bychom chtěli získat hladší trajektorie, je zapotřebí stanovit úlohu polohy ve více bodech. Algoritmus výpočtu úlohy polohy levostranného mechanismu řazení a hodnoty cílové funkce (viz kap. 7.7) napsaný v programovacím jazyku používaném v softwaru MATLAB je přiložen v 11.10.

## 7.7 Tvar cílové funkce

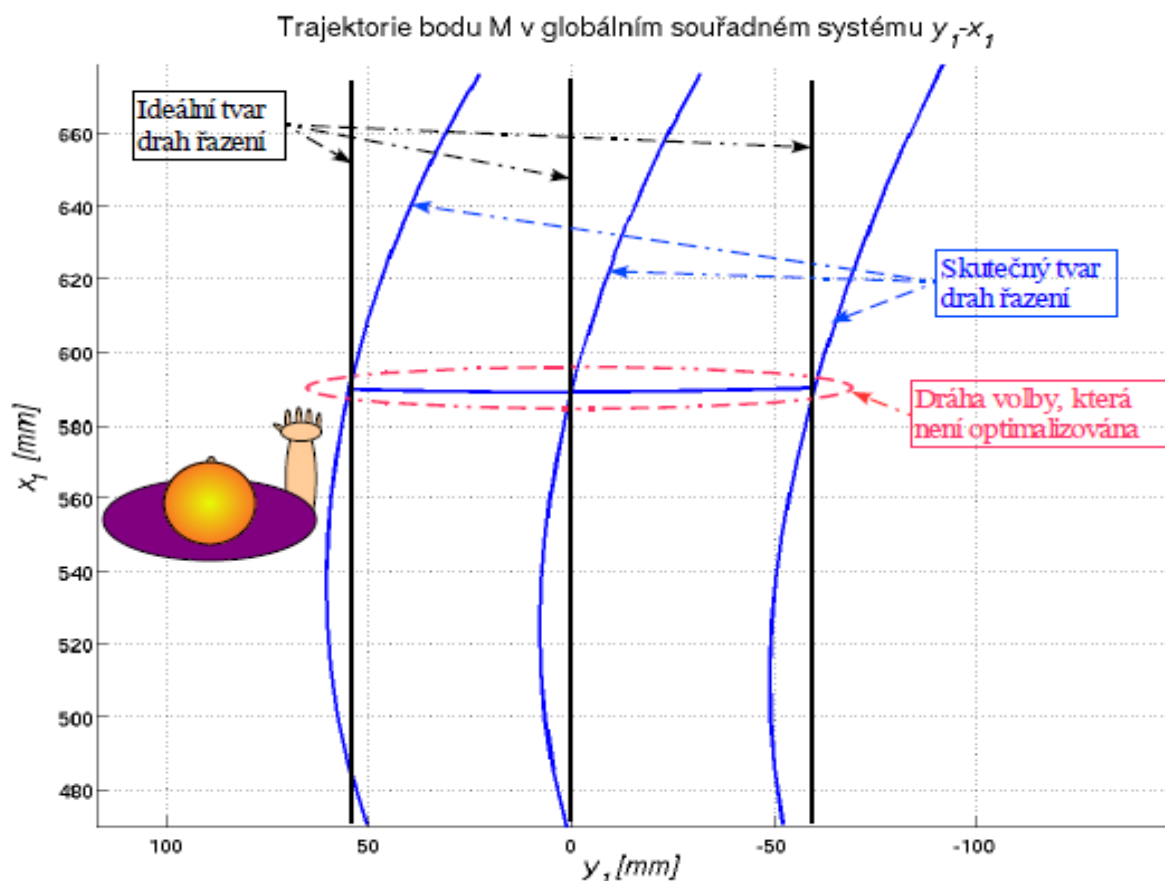
Tvar cílové funkce má při optimalizačním procesu rozhodující vliv na výsledek, proto bylo vytvořeno několik tvarů cílové funkce založených kupříkladu na obsahu ploch trojúhelníků, na

15 Matice  $\mathbf{T}_{68}$  je v tomto tvaru platná pouze v případě, že osa rotační vazby spojující členy 6 a 8 není sklopena pod úhlem, jinak řečeno osa rotační vazby musí být rovnoběžná s osami  $y_6$  a  $y_8$ . V opačném případě je třeba použít pro  $\mathbf{T}_{68}$  vztahu 7.20.



sklonu vyjádřeném ve stupních atd.; nejlépe se osvědčila cílová funkce, jenž bude popsána dále.

Optimalizovány budou pouze ty části drah bodu  $M$  řadicí páky 8, které odpovídají pohybu zvaném řazení, což je pohyb bodu  $M$  ve směru jízdy automobilu, a proto budou dále části drah, které budou optimalizovány, nazývány dráhami řazení. Pohyb volby (pohyb v směru kolmém ke směru jízdy vozu) není při optimalizačním procesu zohledněn. Jak již bylo zmíněno výše, účelem je dráhy řazení napřímit a jejich sklon upravit tak, aby byly rovnoběžné se směrem jízdy automobilu. Ideální tvar drah řazení a příklad skutečného tvaru drah řazení je zobrazen v obr. 7.11.



Obrázek 7.11: Ideální a skutečný tvar drah řazení

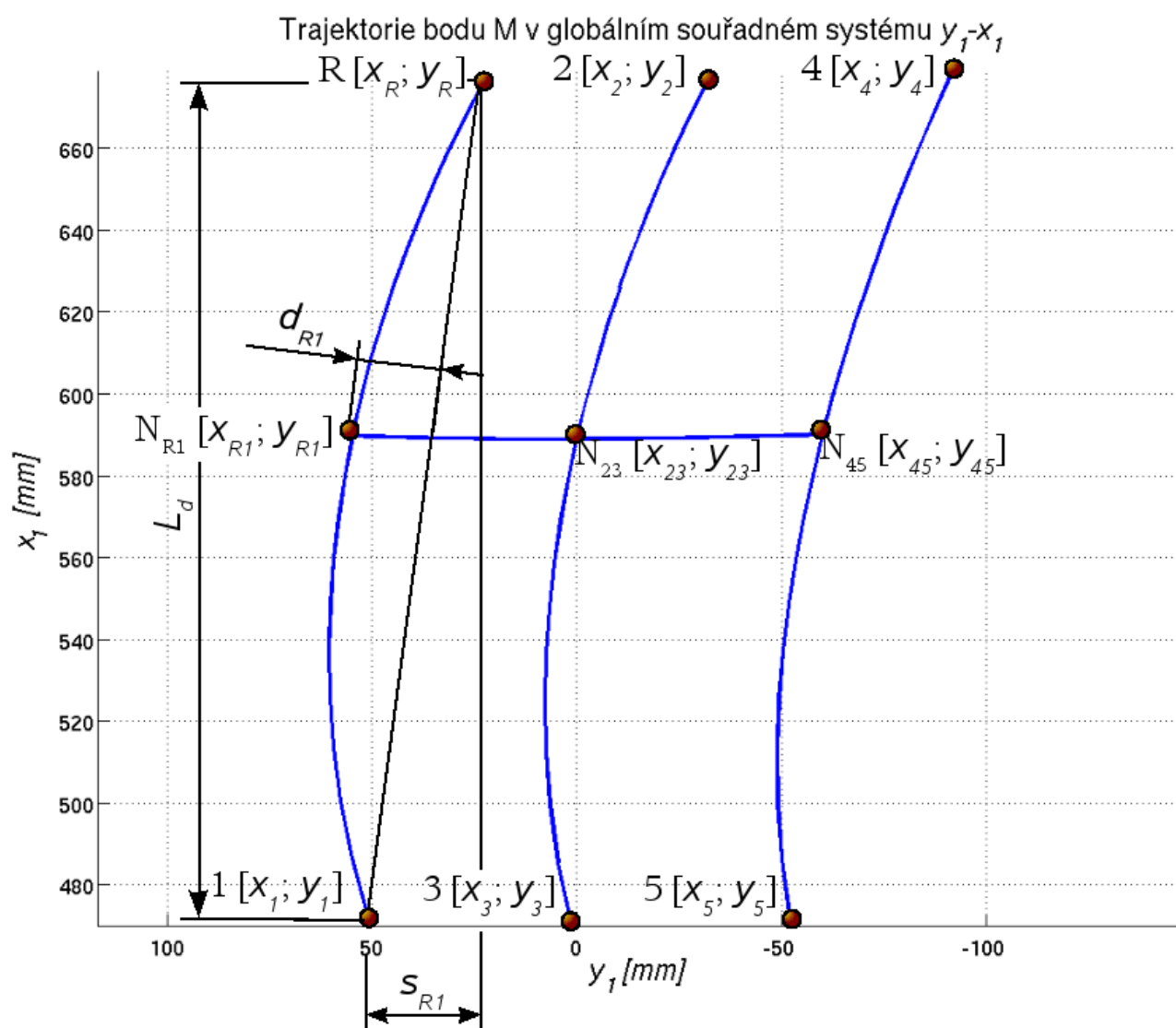
Při návrhu cílové funkce tudíž musíme kvantifikovat dvě základní vlastnosti každé dráhy řazení a to sklon a křivost. Cílová funkce by dále měla být nejlépe kvadratická, protože optimalizační metoda BFGS využívá lokálního kvadratického modelu a pomocí kvadrátů též zajistíme, aby funkce byla kladná. Dále by bylo vhodné vztáhnout obě vlastnosti k délce drah  $L_d$ , viz. obrázek níže. Za předpokladu, že délky drah jsou pro všechny dráhy řazení jen málo odlišné, bude stačit počítat  $L_d$  pouze pro první dráhu

$$L_d = x_R - x_I. \quad (7.13)$$

Cílová funkce  $F$  bude vyjádřena pomocí funkcí  $G_{ij}$ , jenž kvantifikují vlastnosti jednotlivých drah.

### 7.7.1 Funkce kvantifikující vlastnosti jednotlivých drah řazení

Cílová funkce  $F$  je bezrozměrná. V obr. 7.12 jsou označeny souřadnice bodů, které odpovídají rychlostním stupňům a bodům volby, a dále jsou okótovány geometrické charakteristiky drah, které slouží k určení hodnot cílové funkce.



Obrázek 7.12: Označení bodů důležitých při vyčíslování cílové funkce

#### Funkce pro dráhu řazení R1

Sklon dráhy R1 (dráhy z bodu R do 1) je vyjádřen pomocí rozdílu  $y$ -ových souřadnic bodů R a 1 a je označen  $s_{R1}$

$$s_{R1} = y_R - y_1. \quad (7.14)$$

Křivost dráhy R1 je vyjádřena jako  $d_{RI}$  vzdálenost přímky procházející body R a 1 od bodu  $N_{R1}$ . Dle [44] má rovnice přímky procházející body R a 1 následující tvar

$$a_{RI} \cdot x + b_{RI} \cdot y + c_{RI} = 0, \quad (7.15)$$

kde  $a_{RI} = y_R - y_I$ ,  $b_{RI} = x_I - x_R$  a  $c_{RI} = x_R \cdot (y_I - y_R) - y_R \cdot (x_I - x_R)$ ,

pak vzdálenost  $d_{RI}$  bude

$$d_{RI} = \frac{a_{RI} \cdot x_{RI} + b_{RI} \cdot y_{RI} + c_{RI}}{\sqrt{a_{RI}^2 + b_{RI}^2}}. \quad (7.16)$$

Hodnotu funkce  $G_{RI}$  dráhy R1 můžeme nyní vyjádřit jako

$$G_{RI} = \left( \frac{w_s \cdot s_{RI}}{L_d} \right)^2 + \left( \frac{w_d \cdot d_{RI}}{L_d} \right)^2, \quad (7.17)$$

kde skaláry, které nabývají hodnot  $w_s, w_d \in \langle 0; 1 \rangle$ , jsou váhové faktory; je vhodné volit je tak, aby platilo

$$w_s + w_d = 1. \quad (7.18)$$

Těmito faktory lze zesilovat či zeslabovat dopad jedné za dvou stanovených vlastností; například položíme-li  $w_s = 1$  a  $w_d = 0$ , bude optimalizován pouze sklon drah řazení. Pokud nebude uvedeno jinak, budou váhové faktory rovny  $w_s = w_d = 0,5$ .

### Funkce pro dráhu řazení 23

Tvar funkce získáme analogicky záměnou indexů ve vztazích z minulé podkapitoly. Sklon dráhy 23 bude

$$s_{23} = y_2 - y_3, \quad s_{23} = y_2 - y_3. \quad (7.19)$$

Vzdálenost přímky procházející body 2 a 3 od bodu  $N_{23}$  se vypočte

$$d_{23} = \frac{a_{23} \cdot x_{23} + b_{23} \cdot y_{23} + c_{23}}{\sqrt{a_{23}^2 + b_{23}^2}}, \quad (7.20)$$

$$\text{kde } a_{23} = y_2 - y_3, \quad b_{23} = x_3 - x_2 \quad \text{a} \quad c_{23} = x_2 \cdot (y_3 - y_2) - y_2 \cdot (x_3 - x_2).$$

Hodnota funkce  $G_{23}$  dráhy 23 bude potom

$$G_{23} = \left( \frac{w_s \cdot s_{23}}{L_d} \right)^2 + \left( \frac{w_d \cdot d_{23}}{L_d} \right)^2. \quad (7.21)$$

### Funkce pro dráhu řazení 45

Tvar funkce získáme opět analogicky záměnou indexů ve vztazích z minulé podkapitoly. Sklon dráhy 45 bude

$$s_{45} = y_4 - y_5. \quad (7.22)$$

Vzdálenost přímky procházející body 4 a 5 od bodu  $N_{45}$  se vypočte

$$d_{45} = \frac{a_{45} \cdot x_{45} + b_{45} \cdot y_{45} + c_{45}}{\sqrt{a_{45}^2 + b_{45}^2}}, \quad (7.23)$$

$$\text{kde } a_{45} = y_4 - y_5, \quad b_{45} = x_5 - x_4 \quad \text{a} \quad c_{45} = x_4 \cdot (y_5 - y_4) - y_4 \cdot (x_5 - x_4).$$

Hodnota funkce  $G_{45}$  dráhy 45 bude potom

$$G_{45} = \left( \frac{w_s \cdot s_{45}}{L_d} \right)^2 + \left( \frac{w_d \cdot d_{45}}{L_d} \right)^2. \quad 16 \quad (7.24)$$

### 7.7.2 Cílová funkce

Cílová funkce se vypočte jako součet funkcí  $G_{ij}$  jednotlivých drah řazení

$$F = G_{Rl} + G_{23} + G_{45}. \quad (7.25)$$

## 7.8 Rozměry, dráhy řazení a citlivostní analýza levostranného mechanismu řazení

Rozměry mechanismu byly naměřeny na prostorovém počítačovém modelu levostranného mechanismu v programovém prostředí Pro/ENGINEER, viz obr. 7.4 a 7.5, který byl dán k dispozici výrobcem nákladního vozu. Všechny naměřené rozměry jsou uvedeny v obrázku.

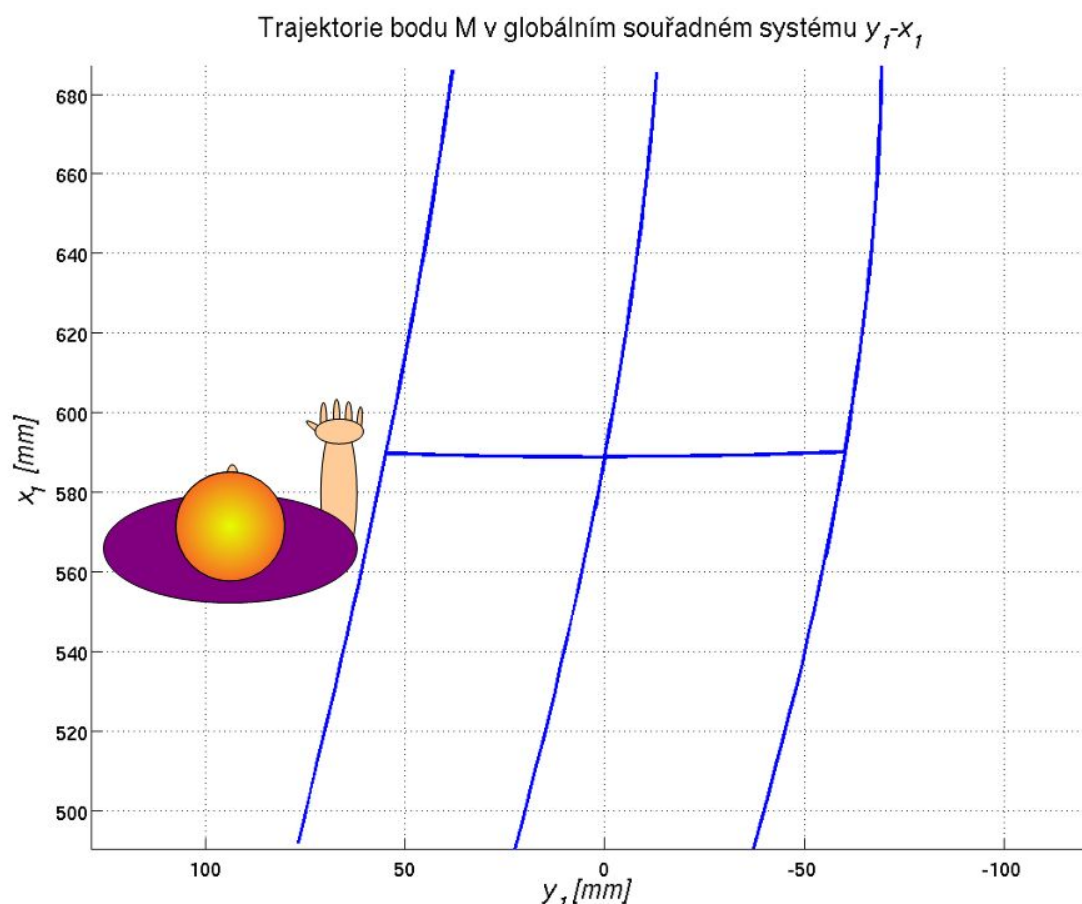
V následujícím obr. 7.14 je vykreslena trajektorie bodu M rukojeti řadicí páky 8 v globálním souřadném systému 1 pro levostranný mechanismus s rozměry uvedenými na obr. 7.13 a s hodnotou úhlu  $\psi = 0$  (viz. závěr této kapitoly). Hodnota cílové funkce  $F$  je v tomto případě

$$F = 258 \cdot 10^{-4}. \quad (7.26)$$

16 : Cílová funkce  $F$  je bezrozměrná.

Obrázek 7.13: Hodnoty rozměrů levostranného mechanismu řazení

Již na první pohled je vidět, že dráhy řazení mají daleko od optima; řidiči nákladního vozu by se řazení jistě nezdálo být přirozené. Obtížně by si na něj zvykal, zvláště v případě častého přechodu z jiného typu automobilu, takže je třeba přejít k optimalizaci pomocí vybraných ladicích parametrů.



Obrázek 7.14: Dráhy řazení neoptimalizovaného levostranného mechanismu řazení

Před započítáním optimalizačního procesu je třeba zvolit ladicí parametry (rozměry mechanismu), které budou optimalizovány. Tyto parametry byly vybrány pomocí citlivostní analýzy, která spočívala v tom, že bylo sledováno, jaký vliv má změna daného rozměru na tvar drah řazení respektive na hodnotu cílové funkce. Citlivostní analýza pro levostranný mechanismus není přiložena z důvodu omezeného rozsahu práce a také proto, že podrobná citlivostní analýza (viz. kap. 8.8) byla provedena pro pravostranný mechanismus. Jako vhodné ladicí parametry byly vybrány rozměry mechanismu  $L_{4s}$  a  $L_{6s}$ , posledním nalezeným vhodným ladicím parametrem je naklopení rotační vazby, která přenáší pohyb řadicí páky 8 na člen 6, o úhel  $\psi$ , viz obr. 7.13. Změna parametru  $\psi$  je z výrobního hlediska nejnáročnější, proto bude zvlášť provedena optimalizace se dvěma ladicími parametry  $L_{4s}$  a  $L_{6s}$ , a se třemi  $L_{4s}$ ,  $L_{6s}$  a  $\psi$ . V případě, že bude uvažován i parametr  $\psi$ , je třeba změnit tvar transformační matice  $T_{68}$ , který byl uveden v 7.9, na tvar

$$T_{68} = T_x(L_{6r}) \cdot T_z(L_{6h}) \cdot R_x(\psi) \cdot R_y(\alpha_{15}) \cdot R_x(-\psi) \cdot T_x(L_{8r}) \cdot T_z(-L_{8s}). \quad (7.27)$$

## 7.9 Optimalizace bez omezení levostranného mechanismu řazení

### 7.9.1 Dvourozměrná optimalizace parametrů $L_{4s}$ a $L_{6s}$ metodou simplexů

Prvky vektoru ladicích parametrů budou uspořádány dle schématu

$$\mathbf{x}_i^{(j)} = [L_{4s}; L_{6s}], \quad (7.28)$$

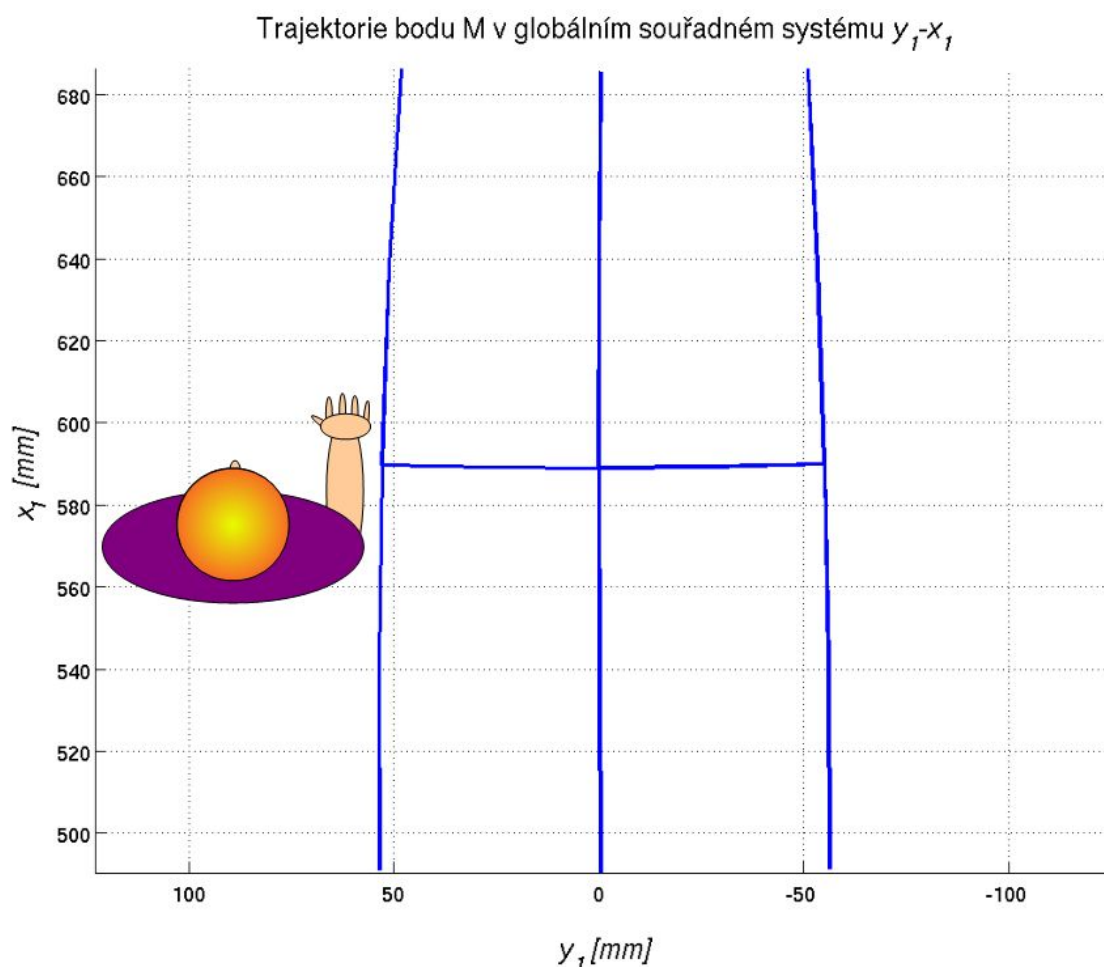
kde  $i = 1, 2, 3$  je pořadové číslo vrcholu simplexu a horní index  $j$  značí iteraci.

Metoda simplexů byla aplikována na levostranný mechanismus řazení s hodnotou

$$\varepsilon_s = 10^{-3} \quad (7.29)$$

(sloužící k ukončení algoritmu), tři vrcholy počátečního simplexu, kterým pro rovinu je trojúhelník, měly následující souřadnice

$$\mathbf{x}_1^{(0)} = [262; 82] \text{ mm}, \quad \mathbf{x}_2^{(0)} = [242; 82] \text{ mm} \quad \text{a} \quad \mathbf{x}_3^{(0)} = [252; 92] \text{ mm}. \quad (7.30)$$



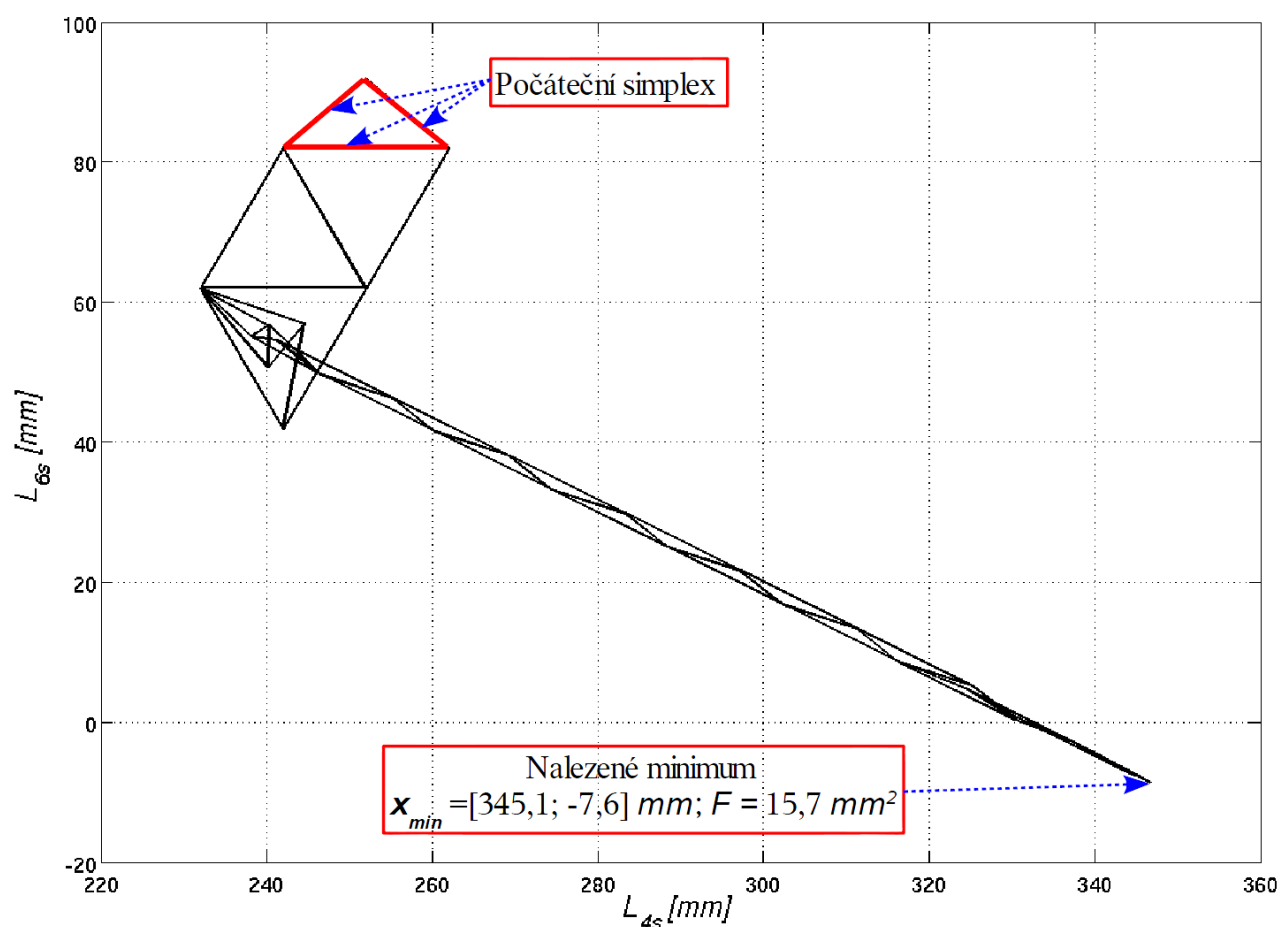
Obrázek 7.15: Dráhy řazení získané dvourozměrnou optimalizací metodou simplexů

Hodnoty ladicích parametrů vypočtených metodou simplexů, hodnota cílové funkce v

tomto bodu a další údaje o průběhu optimalizace jsou shrnuty v tabulce. Průběh optimalizace je znázorněn v obr. 7.16 a v obr. 7.15 je vykreslen tvar drah řazení pro ladící parametry nalezené metodou simplexů.

Tabulka 7.1: Informace o průběhu dvourozměrné optimalizace metodou simplexů

Optimalizační metoda	Počet vyčíslení cílové funkce	Počet iterací	Výsledný vektor ladících parametrů $\mathbf{x}_{min} = [L_{4s}; L_{6s}]$	Hodnota cílové funkce $F_{min}$	Výpočtový čas [minuty:sekundy]
Metoda simplexů	97	49	$[345,1; -7,6] \text{ mm}^{17}$	$4 \cdot 10^{-4}$	0:22



Obrázek 7.16: Průběh dvourozměrné optimalizace metodou simplexů

<sup>17</sup> Záporná hodnota rozměru  $L_{6s} = -7,6 \text{ mm}$  znamená, že rozměr musí být vyneseno opačným směrem než je zakótována na obr. 7.13.



### 7.9.2 Dvourozměrná optimalizace parametrů $L_{4s}$ a $L_{6s}$ kombinovanou metodou BFGS

Počáteční vektor ladicích parametrů pro kombinovanou metodu BFGS byl následující

$$\mathbf{x}^{(1)} = [252; 82] \text{ mm}, \quad (7.31)$$

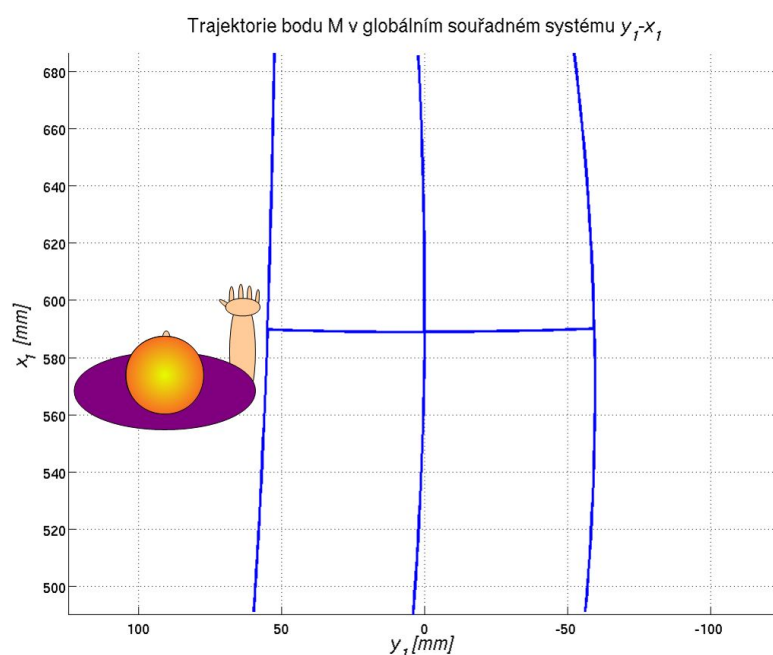
kde pořadí prvků vektoru je  $\mathbf{x}^{(1)} = [L_{4s}; L_{6s}]$ . Hodnota sloužící k ukončení algoritmu byla zvolena

$$\varepsilon_{BFGS} = 10^{-3}. \quad (7.32)$$

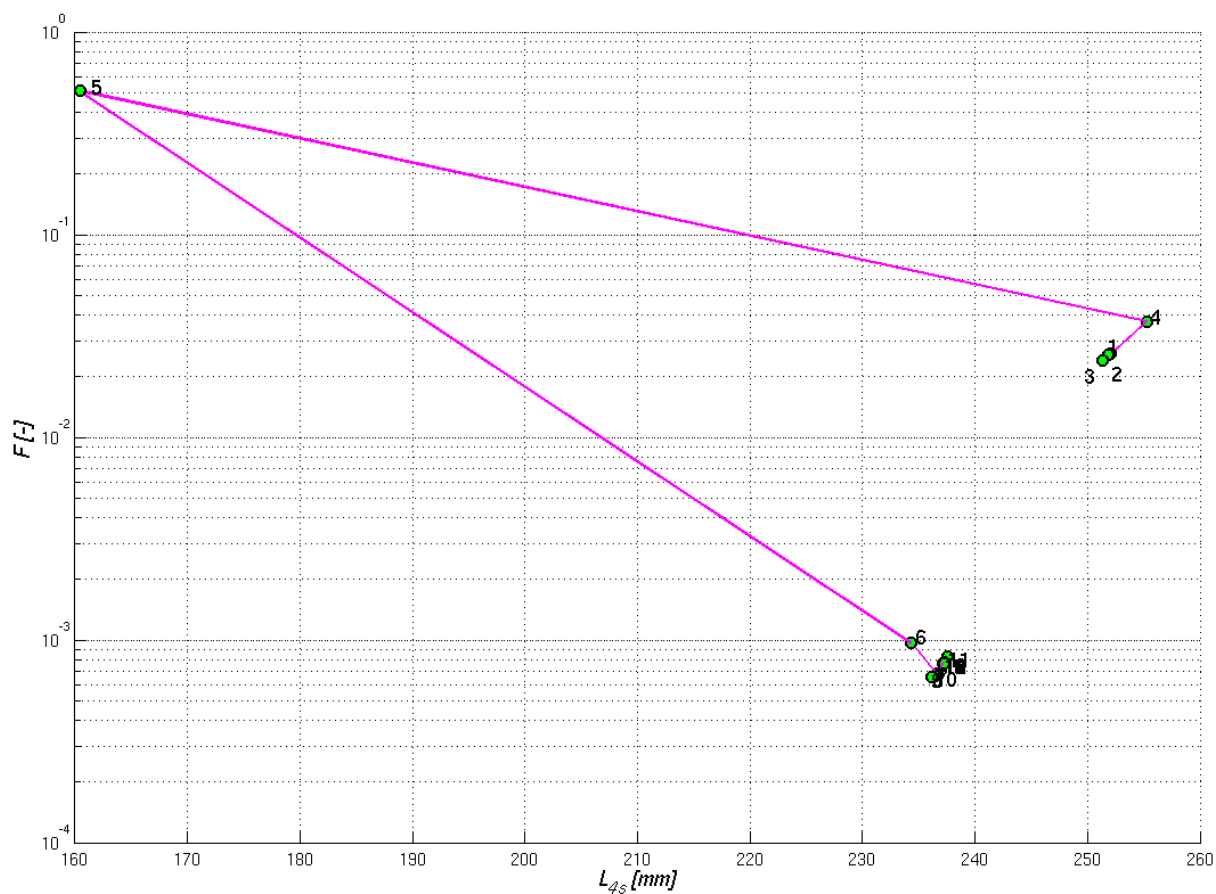
Výsledky jsou opět shrnuty v tabulce. Tvar drah řazení pro minimum nalezené pomocí této metody je vykreslen v obr. 7.17. V obr. 7.18 a 7.19 je zobrazen průběh optimalizace kombinovanou BFGS metodou s hodnotami cílové funkce na  $y$ -ové ose a hodnotou ladicího parametru na  $x$ -ové ose, čísla bodů odpovídají iteraci.

Tabulka 7.2: Informace o dvourozměrné optimalizaci kombinovanou metodou BFGS

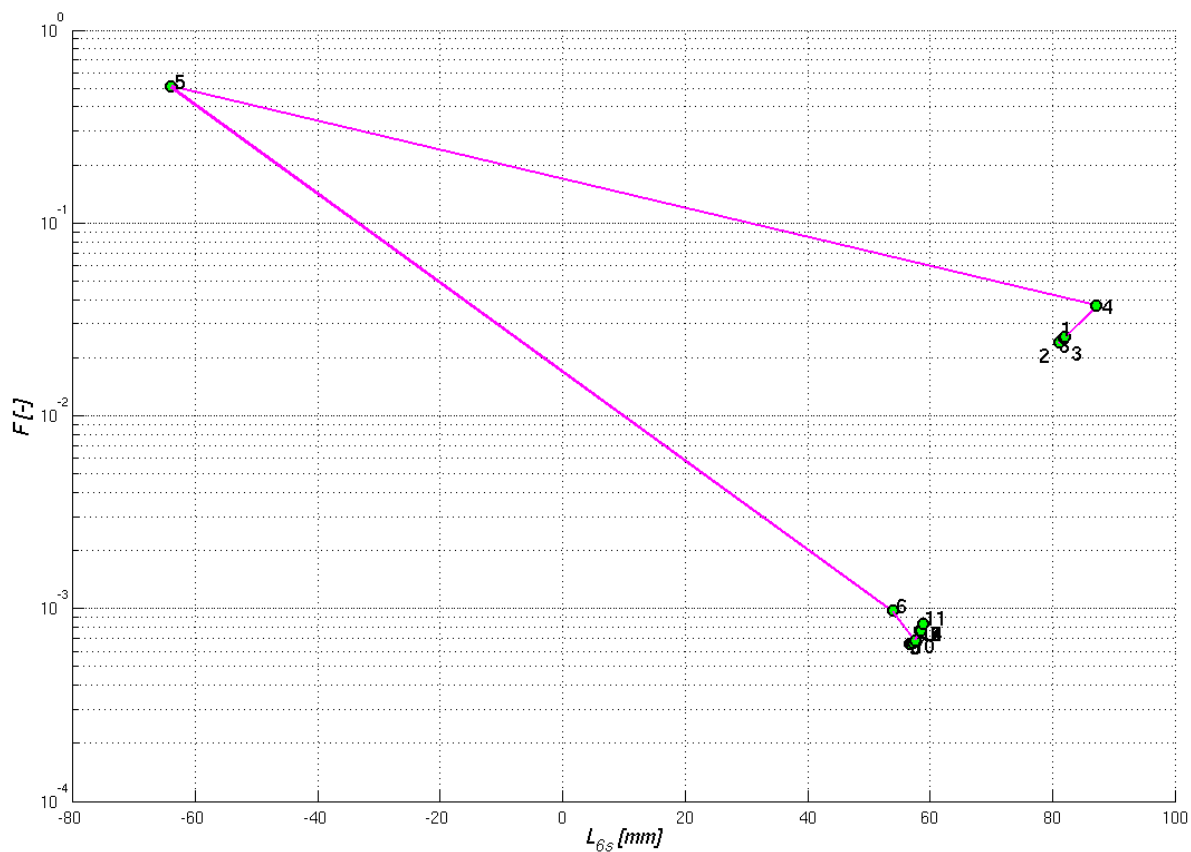
Optimalizační metoda	Počet vyčíslení cílové funkce	Počet iterací	Výsledný vektor ladicích parametrů $\mathbf{x}_{min} = [L_{4s}; L_{6s}]$	Hodnota cílové funkce $F_{min}$	Výpočtový čas [minuty:sekundy]
Kombinovaná metoda BFGS ( $p = 10$ )	50	19	$[237,2; 58,4] \text{ mm}$	$7,7 \cdot 10^{-4}$	0:10



Obrázek 7.17: Dráhy řazení získané dvourozměrnou optimalizací kombinovanou metodou BFGS



Obrázek 7.18: Průběh dvourozměrné optimalizace kombinovanou metodou BFGS



Obrázek 7.19: Průběh dvourozměrné optimalizace kombinovanou metodou BFGS

### 7.9.3 Trojrozměrná optimalizace parametrů $L_{4s}$ , $L_{6s}$ a $\psi$ metodou simplexů

V případě trojrozměrné optimalizace metodou Nelder-Mead se ukázalo vhodnější nastavit hodnotu váhových faktorů  $w_s = 0,3$  a  $w_d = 0,7$ , aby byl kladen větší důraz na křivost dráhy, při rovnajících se váhových faktorech byly výsledné trajektorie příliš zakřivené. Pro tyto hodnoty váhových faktorů je hodnota cílové funkce na pro původní rozměry mechanismu

$$F = 100 \cdot 10^{-4}. \quad (7.33)$$

Hodnota ukončující algoritmus byla zvolena

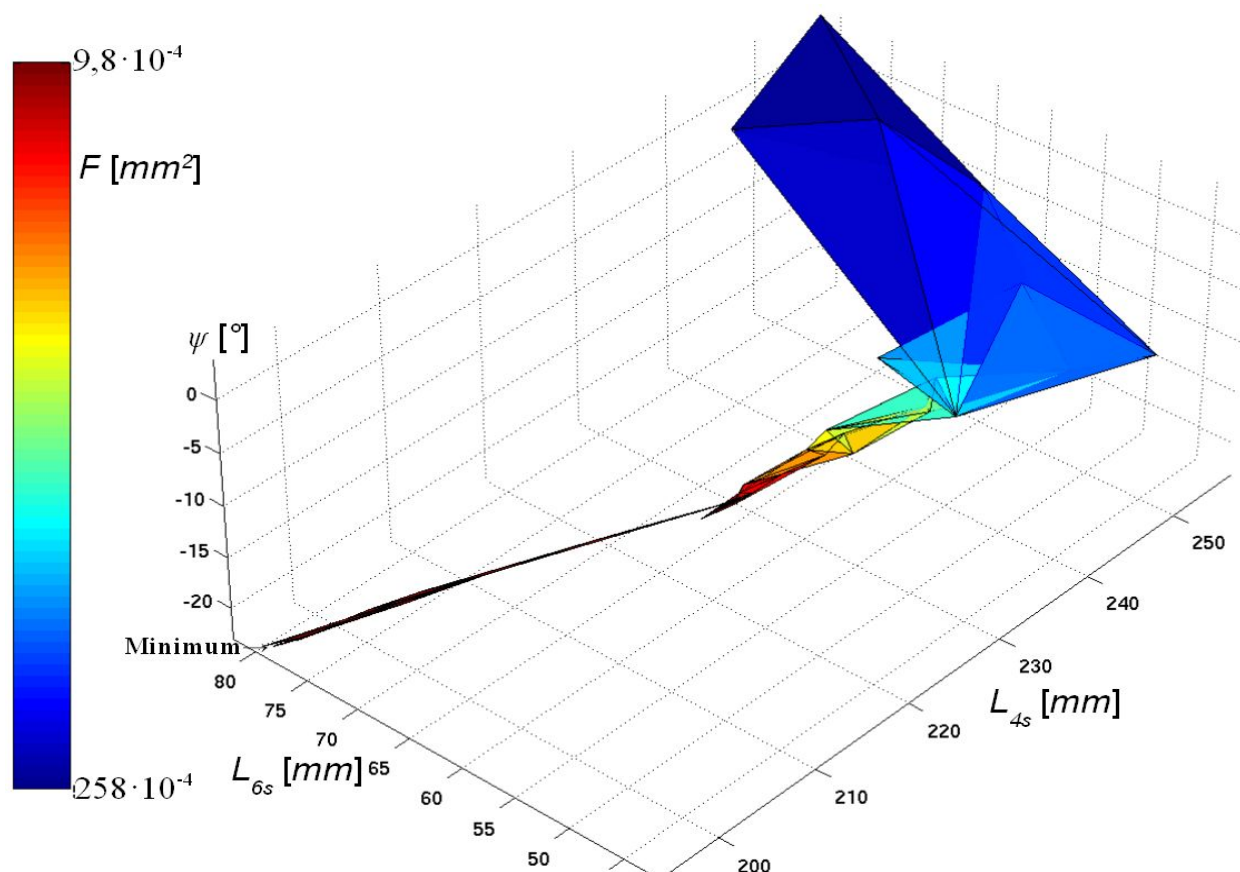
$$\varepsilon_s = 10^{-3}. \quad (7.34)$$

Simplex pro třírozměrnou optimalizaci nabývá tvaru čtyřstěnu, jehož počáteční vrcholy byly zadány

$$\begin{aligned} \mathbf{x}_1^{(0)} &= [252 \text{ mm}; 72 \text{ mm}; 0^\circ], & \mathbf{x}_2^{(0)} &= [257 \text{ mm}; 82 \text{ mm}; 3^\circ], \\ \mathbf{x}_3^{(0)} &= [247 \text{ mm}; 82 \text{ mm}; -3^\circ] & \text{a } \mathbf{x}_4^{(0)} &= [252 \text{ mm}; 87 \text{ mm}; 1^\circ], \end{aligned} \quad (7.35)$$

kde prvky vektoru ladicích parametrů jsou uspořádány dle následujícího schématu

$$\mathbf{x}_i^{(j)} = [L_{4s}; L_{6s}; \psi]. \quad (7.36)$$



Obrázek 7.20: Průběh třírozměrné optimalizace metodou simplexů; simplexem má tvar čtyřstěnu

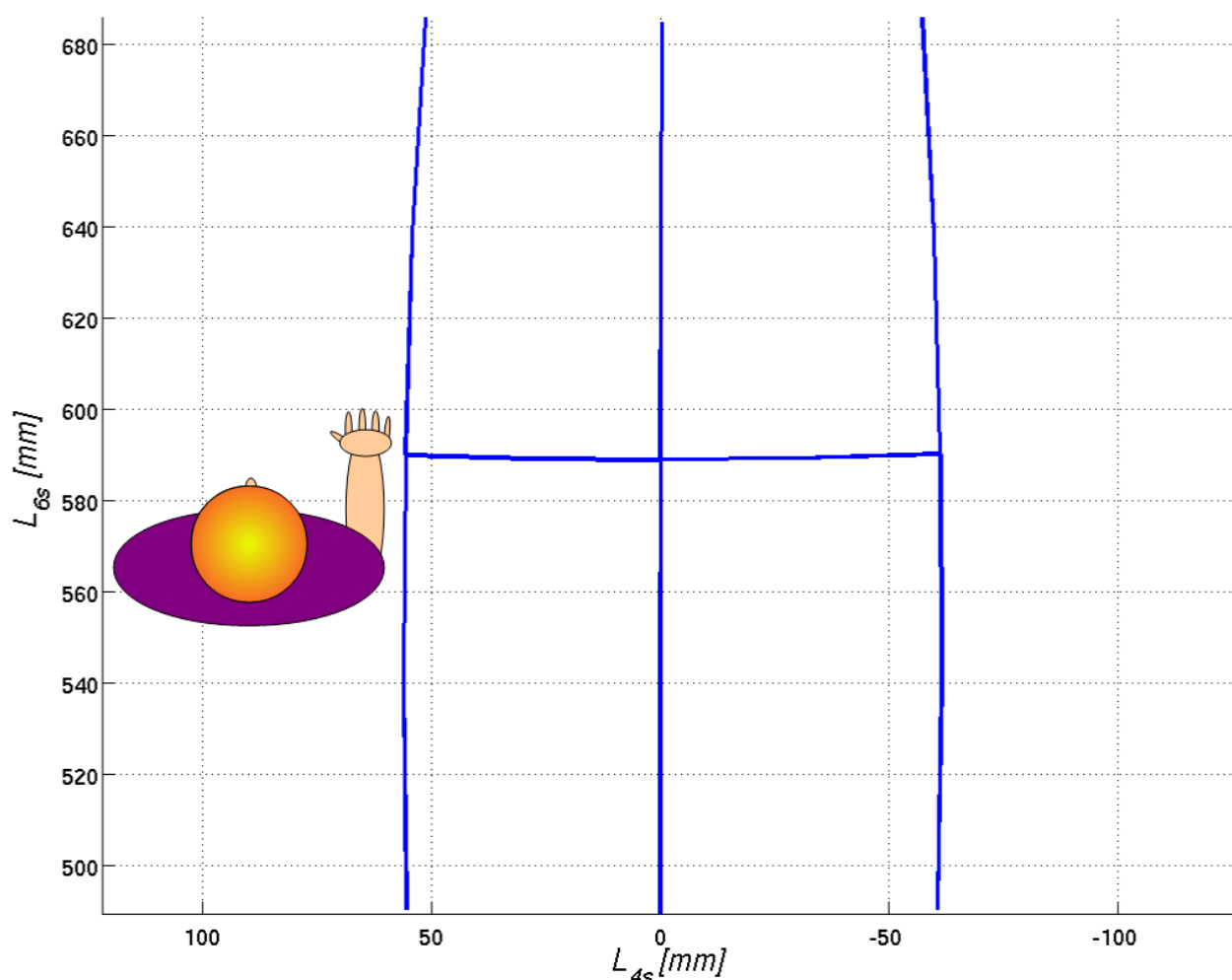
V následujícím obr. 7.20 je vykreslen průběh třírozměrné optimalizace metodou simplexů, na osách jsou vyneseny hodnoty ladicích parametrů, do grafu byly postupně zakreslovány čtyřstěny, jejichž zbarvení odpovídá hodnotě cílové funkce; počáteční simplex je vykreslen modrou barvou a simplex, který je již blízko minima, přechází v barvu červenou.

Výsledky procesu optimalizace jsou opět shrnuty v tab. 7.3 a tvar drah řazení je uveden v následujícím obr. 7.21.

Tabulka 7.3: Informace o třírozměrné optimalizaci metodou simplexů

Optimalizační metoda	Počet vyčíslení cílové funkce	Počet iterací	Výsledný vektor ladicích parametrů $\mathbf{x}_{min} = [L_{4s}; L_{6s}; \psi]$	Hodnota cílové funkce $F_{min}$	Výpočtový čas [minuty:sekundy]
Metoda simplexů	212	115	$[194,8 \text{ mm}; 78,7 \text{ mm}; -22,3^\circ]$	$2,6 \cdot 10^{-4}$	0:48

Trajektorie bodu M v globálním souřadném systému  $y_1-x_1$



Obrázek 7.21: Dráhy řazení získané třírozměrnou optimalizací metodou simplexů

### 7.9.4 Trojrozměrná optimalizace parametrů $L_{4s}$ , $L_{6s}$ a $\psi$ metodou BFGS

Počáteční vektor ladicích parametrů pro kombinovanou metodu BFGS byl

$$\mathbf{x}^{(0)} = [252 \text{ mm}; 82 \text{ mm}; 0^\circ], \quad (7.37)$$

kde pořadí prvků vektoru je  $\mathbf{x}_i^{(j)} = [L_{4s}; L_{6s}; \psi]$ . Hodnota sloužící k ukončení algoritmu byla zadána

$$\varepsilon_{BFGS} = 10^{-3} \quad (7.38)$$

a váhové faktory cílové funkce byly taktéž pozměněny na  $w_s = 0,3$  a  $w_d = 0,7$ .

Tabulka 7.4: Informace o třírozměrné optimalizaci kombinovanou metodou BFGS

Optimalizační metoda	Počet vyčíslení cílové funkce	Počet iterací	Výsledný vektor ladicích parametrů $\mathbf{x}_{min} = [L_{4s}; L_{6s}; \psi]$	Hodnota cílové funkce $F_{min}$	Výpočtový čas [minuty:sekundy]
BFGS – MATLAB přednastavený řešič	166	41	[194,3 mm; 79 mm; -22,4°]	$2,5 \cdot 10^{-4}$	9:21
BFGS – Martinezova hybridní metoda ( $\varepsilon_{NS} = 10^{-11}$ )	96	32	[238.8 mm; 54.2 mm; -14.9°]	$2,1 \cdot 10^{-4}$	0:21
BFGS s linesearch – Martinezova hybridní metoda ( $\varepsilon_{NS} = 10^{-11}$ )	114	22	[238.8 mm; 54.2 mm; -14.9°]	$2,1 \cdot 10^{-4}$	0:26

Výsledný vektor ladicích parametrů je téměř totožný s vektorem získaným metodou simplexů, viz minulá podkapitola; proto není třeba uvádět grafické zobrazení trajektorie bodu  $M$ , které je téměř totožné s obr. 7.21. Získaný vektor ladicích parametrů a další veličiny charakterizující průběh optimalizace jsou uvedeny v tab. 7.4. V této tabulce je shrnuto několik testů. Výpočty v řádek s názvem Gauss-Newton používaly k výpočtu úlohy polohy přednastavený řešič programového prostředí MATLAB, konkrétně Gauss-Newton metodu. V následujícím řádku je využit stejný optimalizační algoritmus BFGS pouze úloha polohy byla řešena navrženým algoritmem Martinezovy hybridní metody. Všimněme si markantního rozdílu ve výpočetním čase. Martinezova hybridní metoda je ve srovnání přednastavenou metodou softwaru MATLAB

$$\frac{t_{MATLAB}}{t_{MARTINEZ}} = \frac{9:21}{0:21} = 27 \text{ krát rychlejší.} \quad (7.39)$$

V posledním řádku tab. 7.4 jsou uvedeny výsledky s implementací „linesearch“ algoritmu „Backtracking-Armijo linesearch“ z [24], viz kap. 63. všimněme si, že výsledky jsou srovnatelné s výpočtem bez použití linesearch pouze výpočetní čas je delší. Proto doporučujeme vynechat linesearch techniku.

### 7.9.5 Trojrozměrná optimalizace bez omezení v multibody softwaru ANSYS Workbench 13.0

Společnost ANSYS vyvíjí již mnoho let prostředí nazvané Workbench, v jehož rámci je dodáván i modul Rigid Dynamics patřící svou charakteristikou do tzv. multibody prostředí, která umožňují simulaci pohybů mechanismů s tuhými členy. Se softwarem ANSYS se na pracovišti, kde vzniká tato práce, pracuje již téměř dvě dekády, a proto je s ním mnoho zkušeností. Z tohoto důvodu byl modul Rigid Dynamics a potažmo modul Goal Driven Optimization, který umožňuje provádět optimalizaci, zvolen jako jeden ze dvou softwarů pro srovnání výsledků optimalizace.

Vzhledem k omezenému rozsahu se zde omezíme na krátké shrnutí postupu a zaměříme se na proces optimalizace, konkrétně na jeho časovou náročnost jak z hlediska přípravy simulace tak výpočtového času. Dále shrneme princip a algoritmy, které využívá optimalizační modul.

#### Parametrický výpočtový model

První bodem optimalizačního procesu ve všech multibody prostředích je vytvoření tzv. parametrického modelu, jehož vstupními parametry jsou zvolené ladicí parametry (v našem případě jsou těmito rozměry  $L_{4s}$ ,  $L_{6s}$  a  $\psi$ ) a výstupním parametrem je hodnota cílové funkce  $F$ . Ladicí parametry byly do modelu nadefinovány jako přírůstky k jejich původním hodnotám z obr. 7.13, takže pokud by výsledkem optimalizace byla 0, tak toto znamená, že hodnota rozměru zůstává stejná jako v původním návrhu.

Tvorba parametrického modelu v modulu Rigid Dynamics byla silně intuitivní, přirozená, jednoduchá a byly k dispozici srozumitelné návody, viz [40] nebo [41]. Časovou náročnost tvorby lze odhadnout na 8 hodin. Nutno upozornit, že geometrický model již byl připraven, takže počítáme pouze úpravy připraveného modelu.

#### Optimalizační proces

Jakmile máme připraven parametrický model, lze přistoupit k optimalizaci. K tomu zde slouží modul Goal Driven Optimization. Optimalizační proces se v tomto modulu skládá ze tří základních bodů: Design of Experiments, Response Surface a Optimization. K modulu je volně

přístupná oficiální literatura, která podrobně shrnuje praktické i teoretické zázemí modulu, viz [42].

## 1. Design of Experiments

Tabulka 7.5: Intervaly ladicích parametrů pro levostranný mechanismus řazení

Ladicí parametr	Dolní mez	Horní mez
$L_{4s}$	192 mm	312 mm
$L_{6s}$	42 mm	122 mm
$\psi$	-25°	30°

V této fázi vybereme ladicí parametry a každému z nich přiřadíme interval, ze kterého mohou hodnoty jednotlivých ladicích parametrů být. Intervaly ladicích parametrů pro levostranný mechanismus řazení byly nastaveny dle tab. 7.5.

Následovně modul navrhne sadu různých kombinací hodnot ladicích parametrů, jednotlivé sady ladicích parametrů jsou nazývány Design Point. Jednoduše lze říci, že modul navrhne síť ladicích parametrů tak, že tyto pokryjí rovnoměrně rozsah jejich hodnot v námi stanovených mezích. Počet Design Point byl v našem případě 15, viz obr. 7.22 kde jsou postupně rozepsány. Modul provede simulaci pro každý tento Design Point a vyčíslí pro něj hodnotu cílové funkce. Výpočet 15 hodnot cílových funkcí trvalo modulu 25 minut. Počáteční vektor ladicích parametrů byl nastaven dle 7.37.

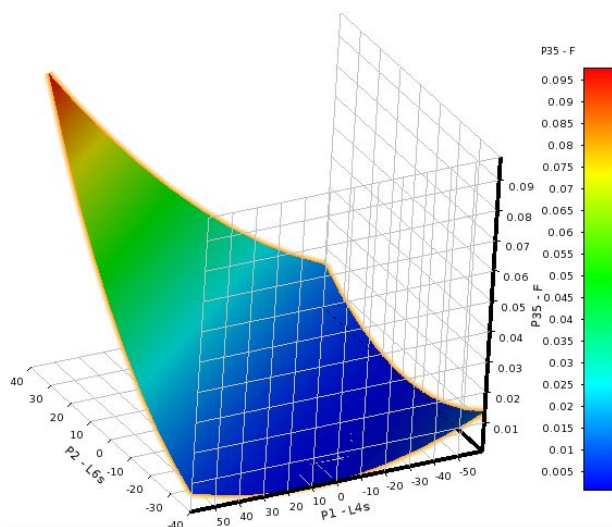
Table of Schematic E2: Design of Experiments					
	A	B	C	D	AK
1	Name	P1 - L4s	P2 - L6s	P37 - psirot	P35 - F
2	1	0	0	2.5	0.0099896
3	2	-60	0	2.5	0.0013824
4	3	60	0	2.5	0.037356
5	4	0	-40	2.5	0.00087427
6	5	0	40	2.5	0.046045
7	6	0	0	-25	0.010109
8	7	0	0	30	0.013851
9	8	-48.782	-32.521	-19.858	0.0042363
10	9	48.782	-32.521	-19.858	0.0072592
11	10	-48.782	32.521	-19.858	0.013328
12	11	48.782	32.521	-19.858	0.06908
13	12	-48.782	-32.521	24.858	0.011311
14	13	48.782	-32.521	24.858	0.009482
15	14	-48.782	32.521	24.858	0.016462
16	15	48.782	32.521	24.858	0.081635

Obrázek 7.22: Síť ladicích parametrů (Design point) a odpovídající hodnoty cílové funkce

## 2. Response Surface

Tato fáze je klíčová pro porozumění principu optimalizace v programovém prostředí ANSYS. Zde modul aproximuje terén cílové funkce  $F$  plochou, která je nazývána Response Surface a dále pracuje již pouze s touto aproximací. Přednastavena je aproximace polynomickou plochou 2.

řádu. Po provedení této aproximace, která trvala 42 minut, lze jednoduše vykreslit mnoho užitečných optimalizačních ukazatelů jako jsou citlivostní grafy apod. Na obr. 7.23 je vykreslen prostorový graf závislosti cílové funkce na ladicích parametrech  $L_{4s}$  a  $L_{6s}$ .



Obrázek 7.23: Prostorový citlivostní graf - závislost cílové funkce na ladicích parametrech  $L_{4s}$  a  $L_{6s}$ .

### 3. Optimization

Poslední fáze celého procesu provede optimalizaci s tím, že již využívá pouze aproximační modely (Response Surface) vytvořené v předchozím kroku a ne parametrický model. Proto se může stát, že hodnota výsledné cílové funkce této optimalizace může být odlišná od skutečné hodnoty pro výsledný vektor ladicích parametrů. Je nezbytné ověřit hodnotu cílové funkce provedením celé simulace. Máme možnost zvolit si následující metody optimalizace s těmito výsledky:

- Screening – neiterativní metoda založená jednoduše na navzorkování specifikované oblasti a následném vyhodnocení vektoru ladicích parametrů s nejmenší hodnotou cílové funkce. Výsledky této metody jsou uvedeny na obr. 7.24. Výpočtový čas byl 4 minuty.
- MOGA – iterativní metoda založená na principu genetických algoritmů, viz [46] nebo [47]. Její výsledky jsou uvedeny na obr. 7.25. Výpočtový čas byl 5 minut.
- NLPQL – iterativní metoda známá jako metoda rozšířených Lagrangiánů. Její výsledky jsou uvedeny na obr. 7.26. Výpočtový čas byl 2 minuty.



Table of Schematic D4: Optimization					
	A	B	C	D	AK
1		P1 - L4s	P2 - L6s	P37 - psirot	P35 - F
2	Optimization Study				
3	Objective	No Objective	No Objective	No Objective	Minimize
4	Target Value				
5	Importance	Default	Default	Default	Default
6	Candidate Points				
7	Candidate A	3.42	-37.382	-9.8833	0.00025863
8	Candidate B	-15.3	-25.429	-13.807	0.00028253
9	Candidate C	-53.46	-6.21	-23.614	0.00028841

Obrázek 7.24: Výsledky optimalizace metodou Screening

Table of Schematic D4: Optimization					
	A	B	C	D	AK
1		P1 - L4s	P2 - L6s	P37 - psirot	P35 - F
2	Optimization Study				
3	Objective	No Objective	No Objective	No Objective	Minimize
4	Target Value				
5	Importance	Default	Default	Default	Default
6	Candidate Points				
7	Candidate A	-0.20271	-35.658	-12.154	0.00025588
8	Candidate B	-2.8724	-33.35	-12.306	0.00025812
9	Candidate C	-8.7642	-30.615	-13.824	0.0002601

Obrázek 7.25: Výsledky optimalizace metodou MOGA

Table of Schematic D4: Optimization					
	A	B	C	D	AK
1		P1 - L4s	P2 - L6s	P37 - psirot	P35 - F
2	Optimization Study				
3	Initial Value	0	0	2.5	
4	Objective	No Objective	No Objective	No Objective	Minimize
5	Target Value				
6	Importance	Default	Default	Default	Default
7	Candidate Points				
8	Candidate A	-13.177	-28.028	-14.38	0.00026297

Obrázek 7.26: Výsledky optimalizace metodou NLPQL

## Shrnutí

Celý optimalizační proces od vytvoření parametrického modelu až po výslednou optimalizaci je z hlediska nároků kladených na znalosti uživatele velice přátelský a lze jej nastavit podle dostupných návodů. Celou úloha byla připravena za 12 hodin. Teorie použitá v optimalizačním modulu je dostupná kvalitně rozepsána v [42]. Zřejmou nevýhodou optimalizace v tomto prostředí je vysoký celkový výpočetní čas, který byl pro náš zkoumaný mechanismus 70 minut.

### 7.9.6 Trojrozměrná optimalizace bez omezení v multibody softwaru ADAMS 2011

Volba dalšího multibody programového prostředí pro srovnání výsledků optimalizace byla vedena tak, aby se jednalo o co možná nejvíce uznávanou špičku v daném oboru. Po provedení rešerše byl jako nejuznávanější multibody software vybrán ADAMS, konkrétně verze 2011.

I toto prostředí obsahuje optimalizační modul, který lze použít k syntéze mechanismů. Jako v předchozí kapitole se omezíme na krátké shrnutí postupu a u procesu optimalizace se zaměříme na jeho časovou náročnost jak z hlediska přípravy simulace tak výpočtového času. Dále shrneme algoritmy, které využívá optimalizační modul.

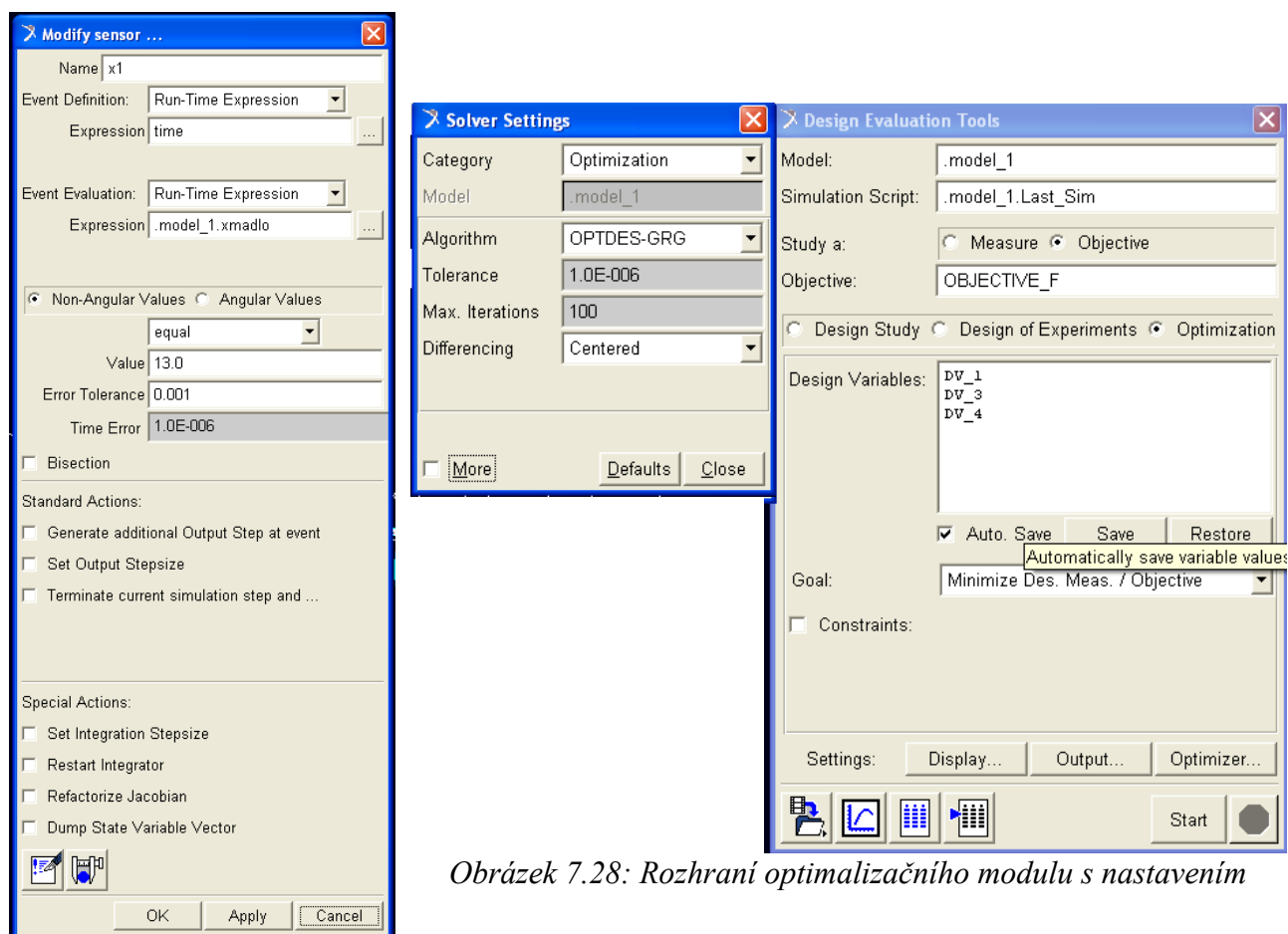
#### Parametrický výpočtový model

Prvním krokem je opět tvorba parametrické simulace pohybu mechanismu. Vstupem do této simulace jsou opět ladicí parametry ( $L_{4s}$ ,  $L_{6s}$  a  $\psi$ ) a výstupem je hodnota cílové funkce  $F$ . Ladicí parametry byly do modelu nadefinovány opět jako přírůstky k jejich původním hodnotám z obr. 7.13.

Zde autor musí poznamenat, že uživatelské rozhraní softwaru je mnohem méně přátelské a intuitivní než u předchozí prezentované prostředí. V nápovědě se obtížně hledají žádané informace a jediný dostupný a současně oficiální návod popisující syntézu mechanismu se zabýval minimalizací reakční síly, viz [43]. V našem případě se však jednalo o syntézu vodícího mechanismu, který se má pohybovat po dané trajektorii. Naneštěstí s těmito typy úloh není mnoho zkušeností ani v konzultační středisku ADAMS pro ČR.

Právě tvorba cílové funkce  $F$ , která je počítána ze souřadnic bodu madla řadicí páky, se ukázala jako značně obtížná. Řešením se ukázalo využití funkce sensor, u který pracuje následovně: když je čas roven dané hodnotě, zapiš hodnotu souřadnice bodu, viz obr. 7.27. Protože se cílová funkce počítá z 9 bodů, z nichž má každý dvě souřadnice, bylo zavedeno 18 sensorů. S těmito sensory se dále pracuje v příkazovém řádku pomocí funkce `senval`, která vyhodnotí daný sensor v čase.

Tvorba parametrického modelu trvala autorovi 2 týdny.



Obrázek 7.28: Rozhraní optimalizačního modulu s nastavením

Obrázek 7.27: Nastavení funkce Sensor pro zapsání  $x$ -ové polohy madla v čase 13 sekund

## Optimalizační modul

Rozhraní optimalizačního modulu je uvedeno na obr. 7.28. Jakmile je připraven parametrický model, je samotná optimalizace a její příprava velice snadná. Prostředí nabízí tři základní metody optimalizace:

- OptDes GRG - gradientní metoda, jejíž největší klad je robustnost.
- OptDes SQP - iterativní metoda nelineární optimalizace. O kterou konkrétní metodu se jedná není blíže specifikováno. Optimalizační proces by měl být rychlejší než předchozí metoda.
- SDI - stochastická metoda, která v každé iteraci náhodně generuje několik sad ladicích parametrů, z nichž vybere nejvhodnějšího kandidáta (nejmenší hodnota cílové funkce).

Dále je možné implementovat vlastní optimalizační algoritmy. V našem případě byla

využita robustní gradientní metoda.

Výpočtový čas optimalizačního procesu byl překvapivě krátký, konkrétně 32 sekund. Výsledná data optimalizačního procesu jsou uvedena v tab. 7.6.

Tabulka 7.6: Výsledná data jednotlivých iterací optimalizačního procesu v softwaru ADAMS 2011

Iter.	$F$ [-]	přírůstek $L_{6s}$ [mm]	přírůstek $L_{4s}$ [mm]	$\psi$ [°]
0	0.00985	0	0	0
1	0.00049	-20.22	-25.29	-0.55
2	0.00024	-21.61	-20.36	-10.14
3	0.00022	-23.13	-22.23	-10.23
4	0.00022	-23.13	-22.23	-10.23
	$F$	$L_{6s}$ [mm]	$L_{4s}$ [mm]	$\psi$ [°]
Výsledný vektor ladicích parametrů	0.00022	58.9	228.9	-10.23°

## Shrnutí

Lze říci, že multibody prostředí ADAMS nabízí zcela opačné vlastnosti vzhledem k předchozímu prostředí ANSYS Workbench. Prostředí ADAMS je méně intuitivní a uživatelsky přívětivé, zato však efektivita algoritmů je vysoká a výpočtové časy velmi nízké.

### 7.9.7 Srovnání výsledků získaných autorem a multibody prostředími – optimalizace bez omezení

Nyní srovnáme výsledky provedené optimalizační úlohy bez omezení různými metodami prezentovanými v kap. 7.9.3, 7.9.4, 7.9.5 a 7.9.6. Výsledná data optimalizačních procesů jsou porovnána v tab. 7.7. Z tabulky je zřejmé, že existuje několik lokálních minim cílové funkce a tudíž terén této funkce bude dosti komplikovaný, viz kap. 4.5. Všimněme si, že každá metoda našla odlišný výsledný vektor ladicích parametrů.

Z pohledu efektivity i časové náročnosti je na tom nejlépe výpočet provedený autorem metodou BFGS. Optimalizace provedená prostředím ADAMS má druhý nejlepší čas i výsledek. Připomeňme však, že příprava optimalizace v tomto softwaru byla značně náročná na uživatele. Z hlediska výpočtového času zcela propadá software ANSYS, který však je potřeba ocenit za velice příjemné a intuitivní ovládání. Metodu simplexů lze pro její robustnost doporučit pro případ, že by metoda BFGS havarovala. Vzhledem k její definici nutně musí být méně efektivní než metoda BFGS.

Tabulka 7.7: Srovnání výsledků optimalizační úlohy bez omezení provedené v kap. 7.9.3, 7.9.4, 7.9.5

Optimalizační metoda	Počet vyčíslení cílové funkce	Počet iterací	Výsledný vektor ladicích parametrů $\mathbf{x}_{min} = [L_{4s}; L_{6s}; \psi]$	Hodnota cílové funkce $F_{min}$	Výpočtový čas [minuty:sekundy]
Metoda simplexů	212	115	[194,8 mm; 78,7 mm; -22,3°]	$2,6 \cdot 10^{-4}$	0:48
BFGS	96	32	[238.8 mm; 54.2 mm; -14.9°]	$2,1 \cdot 10^{-4}$	0:21
ANSYS <sup>18</sup>	-	-	[255.4 mm; 44.6 mm; -9.9°]	$2,6 \cdot 10^{-4}$	70:00
ADAMS	-	-	[228.9 mm; 58.9 mm; -10.23°]	$2,2 \cdot 10^{-4}$	0:32

Je nezbytné připomenout, že takto krátké optimalizační časy autorových výpočtu jsou důsledkem efektivního výpočtu úlohy polohy a tudíž výsledkem výzkum prezentovaného v kap. 5. Pokud bychom použili přednastavené řešiče softwaru MATLAB, tak by dle 7.39 byly výpočty dle 7.39 27-krát pomalejší.

## 7.10 Optimalizace s omezením levostranného mechanismu řazení

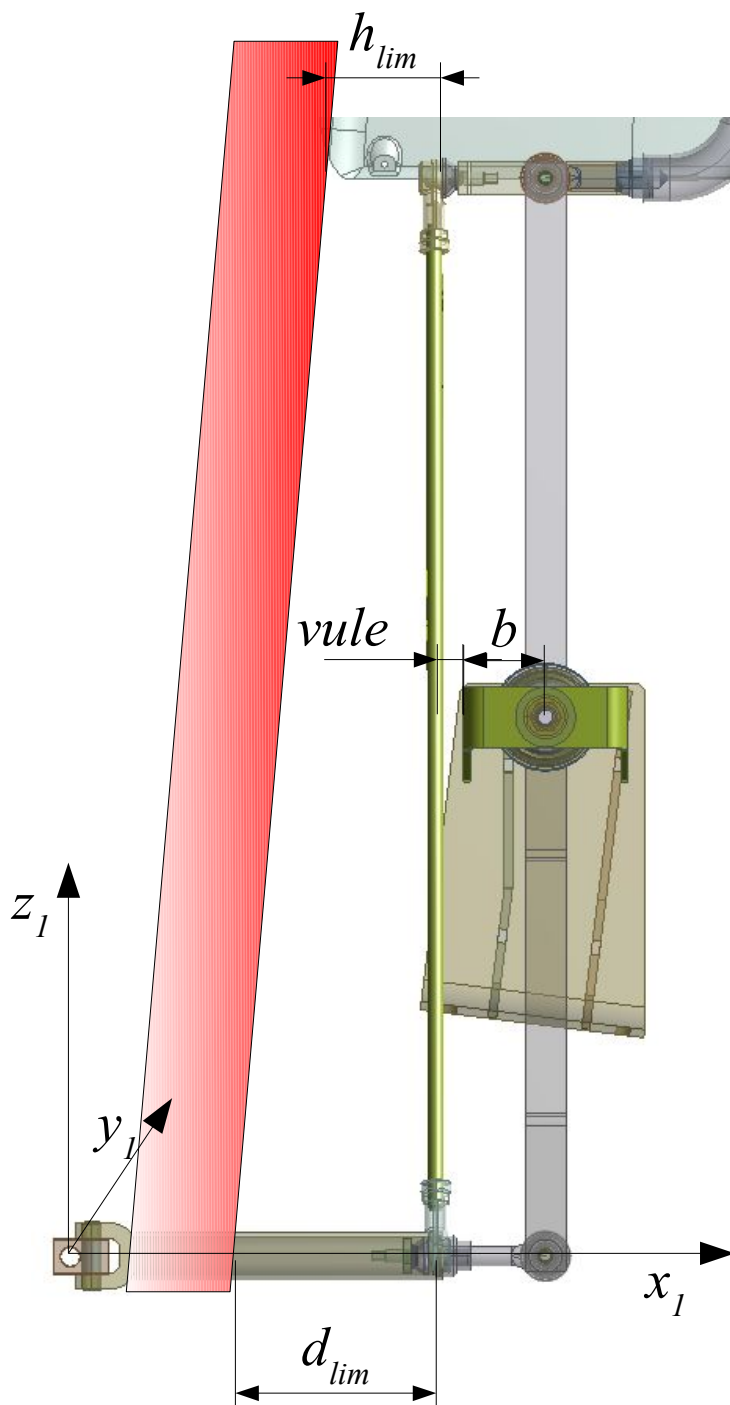
V předchozích kapitolách byla prezentována data získaná optimalizací bez omezujících podmínek, což má za důsledek, že výsledný vektor ladicích parametrů může vést k tomu, že jednotlivé členy mechanismu budou vzájemně v kolizi nebo může docházet ke kolizi s okolním vybavením vozidla, což se v praxi nazývá hledisko zástavby. Chceme-li aby optimalizační proces navrhoval výsledný vektor ladicích parametrů tak, aby tyto nezpůsobovaly kolize, je nutno použít metody optimalizace s omezovacími podmínkami a tyto omezovací podmínky definovat.

Obr. 7.29 bude sloužit jako podklad pro stanovení omezujících podmínek. Tyto podmínky se budou týkat ladicích parametrů  $L_{4s}$  a  $L_{6s}$ . První omezující podmínka vyplývá z toho, že tyč 7 nesmí být v kolizi s ukotvením páky 5. Dále mezi ukotvením a tyčí 7 musí být zachována jistá vůle, aby nedocházelo ke kolizi během pohybu soustavy. Vzhledem k tomu, že ukotvení páky 5 je v polovině, lze zapsat podmínku omezení následovně

$$\frac{-L_4 - x_1 + x_2}{2} + b + vule \leq 0, \quad (7.40)$$

kde  $x_1$  představuje  $L_{4s}$  a  $x_2$  značí  $L_{6s}$ .

<sup>18</sup> Pro tabulku byl vybrán výsledek s nejnižší hodnotou cílové funkce při výběru ze všech metod, tzn. z obr. 7.24, 7.25 a 7.26. Nejlepší výsledek odpovídá metodě Screening z obr. 7.24.



Obrázek 7.29: Návrh zástavby vozu sloužící jako podklad pro návrh omezujících podmínek

Dále je volba ladicích parametrů omezena z důvodů zástavby, která neumožňuje aby poloha sférických kloubů byla volena příliš vzadu, viz obr. 7.29. Další omezující podmínky jsou potom formulovány následovně

$$\begin{aligned} -x_1 + L_{4s \text{ původní}} - d_{\text{lim}} &\leq 0, \\ x_2 - (L_{6s \text{ původní}} + h_{\text{lim}}) &\leq 0, \end{aligned} \quad (7.41)$$

kde  $L_{4s \text{ původní}}$  a  $L_{6s \text{ původní}}$  označují původní rozměry definované v obr. 7.29.

Máme-li definovány podmínky omezení můžeme formulovat optimalizační úlohu jako

$$\mathbf{x}_{min} : \min_{\mathbf{x} \in \mathbb{R}^n} F(\mathbf{x}) \quad \text{omezeno} \quad \left\{ \begin{array}{l} \frac{-L_4 - x_1 + x_2}{2} + b + vule \leq 0, \\ -x_1 + L_{4s \text{ původní}} - d_{lim} \leq 0, \\ x_2 - (L_{6s \text{ původní}} + h_{lim}) \leq 0, \end{array} \right. \quad (7.42)$$

kde  $\mathbf{x}$  je vektor ladicích parametrů, který je uspořádán následně

$$\mathbf{x} = [L_{4s}; L_{6s}; \psi], \quad (7.43)$$

viz obr. 7.29. Hodnoty váhových faktorů byly nastaveny

$$w_s = 0.3 \text{ a } w_d = 0.7. \quad (7.44)$$

Numerické testy prezentované v následujících podkapitolách byly provedeny pro hodnoty parametrů podmínek omezení dle tab. 7.8.

Tabulka 7.8: Hodnoty použité pro omezující podmínky.

Parametr	Hodnota [mm]
$b$	60
$vule$	14
$d_{lim}$	52
$h_{lim}$	18

### 7.10.1 Trojrozměrná optimalizace penalizační metodou

Optimalizační úloha s omezujícími podmínkami definovaná v 4.33 byl řešena metodou penalizace navrženou v kap. 4.4.1. Počáteční vektor ladicích parametrů pro kombinovanou metodu BFGS byl

$$\mathbf{x}^{(0)} = [252 \text{ mm}; 82 \text{ mm}; 0^\circ], \quad (7.45)$$

Parametry algoritmu III byly nastaveny následovně

$$\mu^{(0)} = 0.1, \beta = 4, \nu = 0.001. \quad (7.46)$$

Pro hledání minima penalizované funkce byla využita metoda BFGS respektive navržený algoritmus II s hodnotou sloužící k ukončení algoritmu

$$\varepsilon_{BFGS} = 10^{-3}. \quad (7.47)$$

Pro výpočet úlohy polohy, která je součástí výpočtu úlohy polohy, byla využita Martinezova hybridní metoda s hodnotou ukončující algoritmus

$$\varepsilon_{NS} = 10^{-11}. \quad (7.48)$$

Výsledky algoritmu jsou shrnuty v tab. 7.9.

Tabulka 7.9: Výsledky optimalizace s omezení penalizační metodou pro levostranný mechanismus

Optimalizační metoda	Počet výčíslení cílové funkce	Počet iterací	Výsledný vektor ladicích parametrů $\mathbf{x}_{min} = [L_{4s}; L_{6s}; \psi]$	Hodnota cílové funkce $F_{min}$	Kontrola porušení podmínek omezení	Výpočtový čas [minuty:sekundy]
Penalizační metoda a metoda BFGS	313	1	[238.4 mm; 54.4 mm; -14.9°]	$2.1 \cdot 10^{-4}$	[0 mm, 0 mm, 0 mm]	00:41

### 7.10.2 Trojrozměrná optimalizace provedená v multibody prostředích

Optimalizace s omezením probíhá v multibody softwarech zcela stejným způsobem jako bylo popsáno v kap. 7.9.5 a 7.9.6. Jedinou změnou je, že oproti přibývá implementace podmínek omezení. Porušení podmínek omezení je třeba vyjádřit v parametrickém modelu jako výstupní proměnnou modelu. U obou softwarů lze tyto podmínky definovat podobným způsobem jako cílovou funkci.

Ladicí parametry byly do modelu opět nadefinovány jako přírůstky k jejich původním hodnotám z obr. 7.13, takže pokud by výsledkem optimalizace byla 0, tak toto znamená, že hodnota rozměru zůstává stejná jako v původním návrhu.

Table of Schematic D4: Optimization								
	A	B	C	D	AK	AL	AM	AN
1		P1 - L4s	P2 - L6s	P37 - psirot	P35 - F	P38 - C1	P39 - C2	P40 - Output Parameter
2	Optimization Study							
3	Objective	No Objective	No Objective	No Objective	Minimize	Values <= Target	Values <= Target	Values <= Target
4	Target Value					0	0	0
5	Importance	Default	Default	Default	Default	Default	Default	Default
6	Candidate Points							
7	Candidate A	-23.94	-23.554	-17.436	★★★ 0.00028643	-0.19312	★★ -41.554	★ -28.06
8	Candidate B	-34.74	-16.523	-13.663	★★★ 0.00030306	★ -9.1088	★★ -34.523	-17.26
9	Candidate C	-47.7	-8.085	-14.569	★★★ 0.00031389	★ -19.808	★ -26.085	-4.3

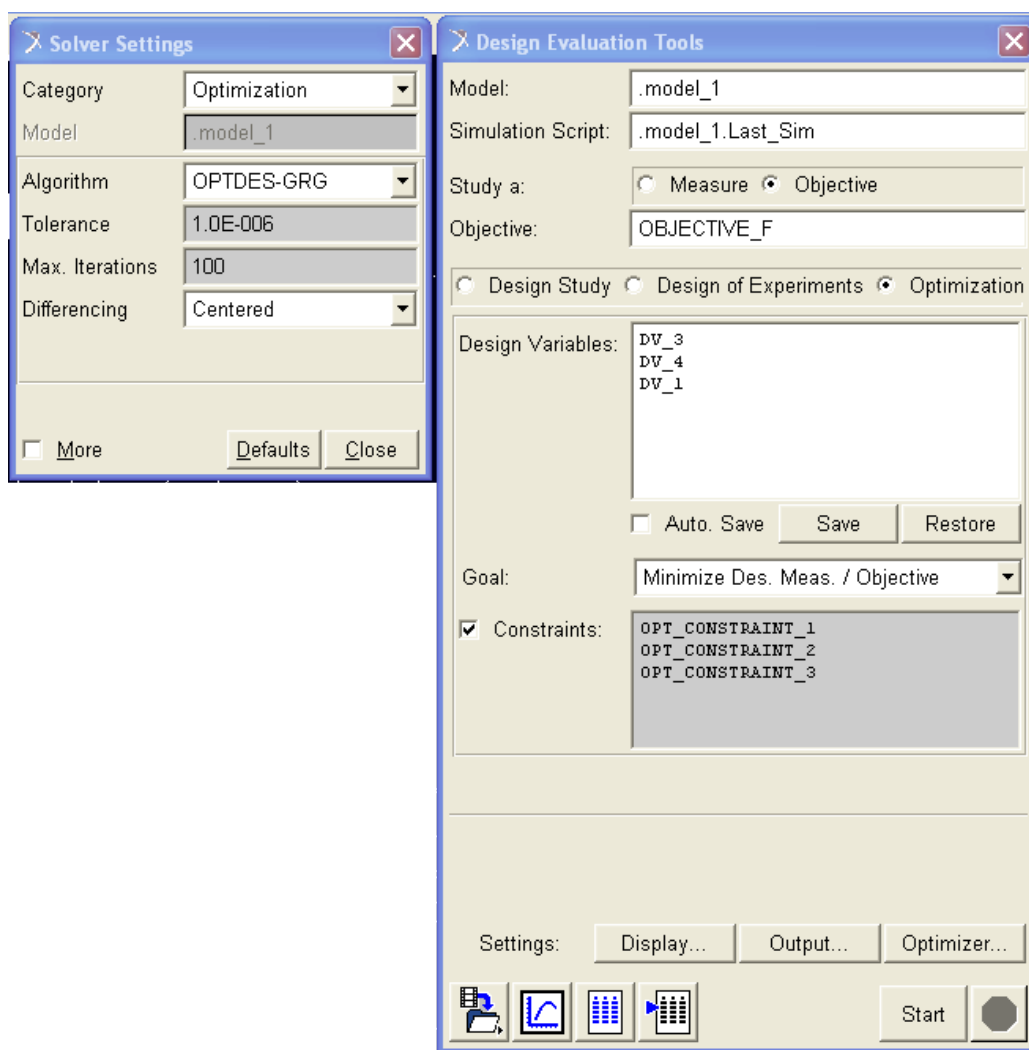
Obrázek 7.30: Nastavení optimalizace – metoda Screening; první tři sloupce - ladicí parametry; čtvrtý sloupec - cílová funkce; zbylé sloupce - podmínky omezení

V prostředí ANSYS podmínky omezení nastavíme v závěrečné fázi optimalizace nazvané Optimization, kde lze volit rovnostní či nerovnostní typ podmínky apod., viz obr. 7.30.



Výsledky získané tímto modulem byly mírně horší než výsledky získané optimalizací provedenou autorem. Prezentována jsou výsledky metody Screening, která dosáhla nejlepších výsledků. Časová náročnost je přibližně stejná byla uvedena v kap. 7.9.5, tj. 70 minut.

U softwaru ADAMS přidáváme podmínky omezení volbou Constraints v nastavení optimalizace dle obr. 7.31. Výsledky získané z prostředí ADAMS jsou totožné s výsledky získanými autorem, viz tab. 7.9. Výpočetní čas byl přibližně 1 minuta.



Obrázek 7.31: Nastavení optimalizace s omezením v prostředí ADAMS; omezující podmínky jsou přidány v oknu Constraints

### 7.10.3 Srovnání výsledků získaných autorem a multibody prostředí – optimalizace bez omezení

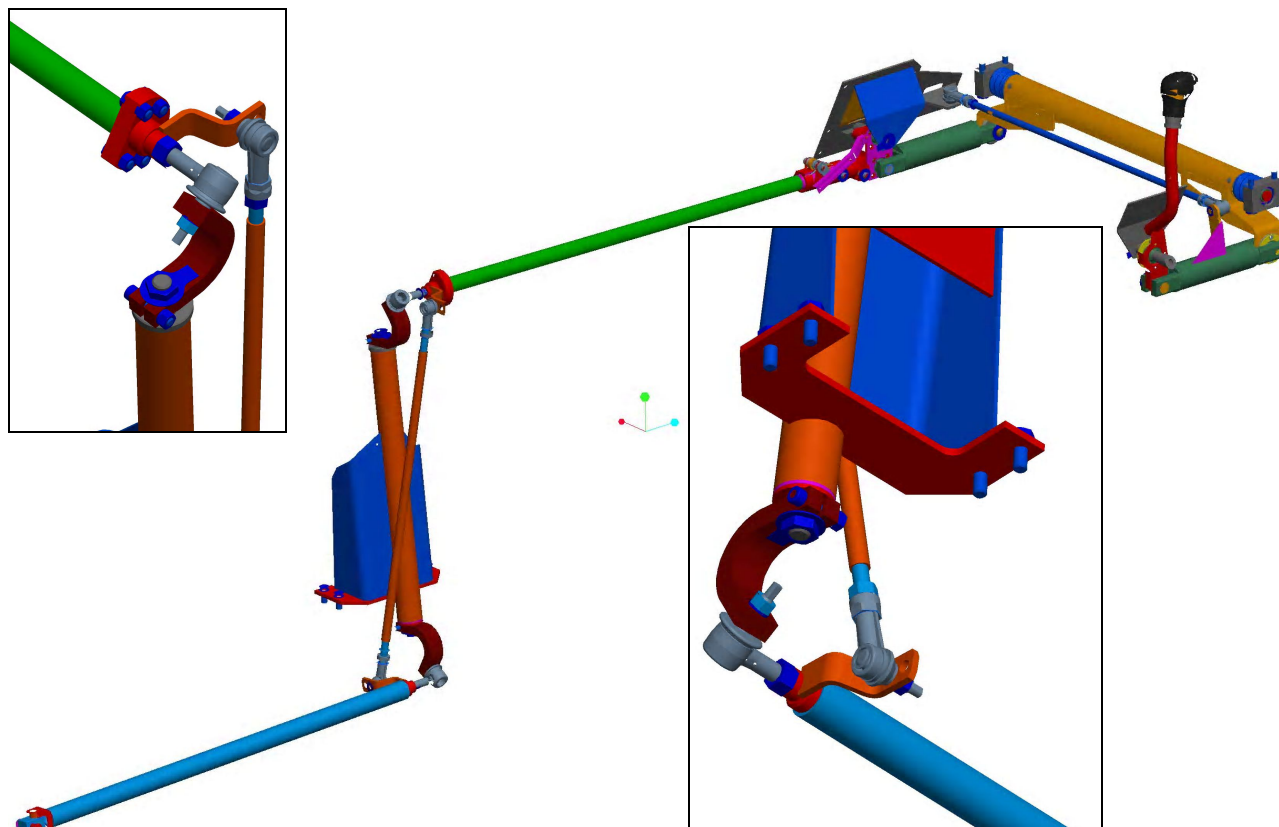
V tab. 7.10 jsou srovnány výsledky optimalizace s omezujícími podmínkami prezentované v kap. 7.10.1 a 7.10.2. Mírně slabší výsledky byly získány z prostředí ANSYS. Výsledky získané autorem metodou penalizace a ze softwaru ADAMS byly srovnatelné. Výpočtový čas je opět zdaleka nejhorší u prostředí ANSYS a nejrychlejší se ukázala penalizační metoda. Hlubší srovnání softwarů je uvedeno v předchozí kap. 7.9.7. Zde platí stejné závěry.

Tabulka 7.10: Srovnání výsledků optimalizační úlohy s omezením

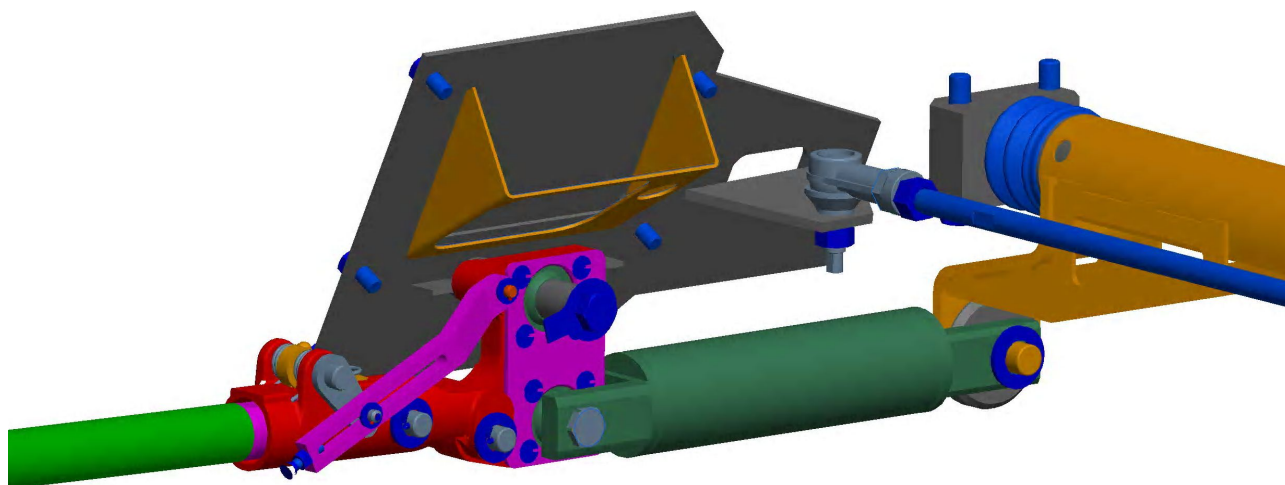
Optimalizační metoda	Počet vyčíslení cílové funkce	Počet iterací	Výsledný vektor ladicích parametrů $\mathbf{x}_{min} = [L_{4s}; L_{6s}; \psi]$	Hodnota cílové funkce $F_{min}$	Kontrola porušení podmínek omezení	Výpočtový čas [minuty:sekundy]
Penalizační metoda a metoda BFGS	313	1	[238.4 mm; 54.4 mm; -14.9°]	$2.1 \cdot 10^{-4}$	[0 mm, 0 mm, 0 mm]	00:41
ANSYS	-	-	[228.1 mm; 58.4 mm; -17.4°]	$2.8 \cdot 10^{-4}$	[0 mm, 0 mm, 0 mm]	70:00
ADAMS	-	4	[238.4 mm; 54.4 mm; -14.9°]	$2.1 \cdot 10^{-4}$	[0 mm, 0 mm, 0 mm]	01:00

## 8 Pravostranný mechanismus řazení

Pro vozidla jezdící po levé straně vozovky se používá pravostranný mechanismus řazení. Vzhledem ke složitosti pravostranného mechanismu řazení byla pro lepší představu o jeho pohybu vytvořena videosekvence, která je přiložena na CD. Prostorový model mechanismu, který byl dodán od výrobce, je vyobrazen na následujících obr. 8.1, 8.2 a 8.3.

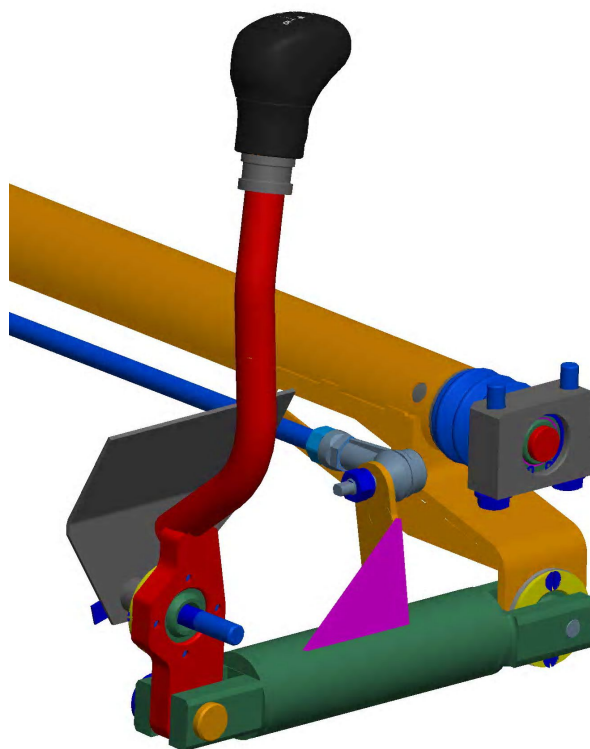


Obrázek 8.1: Prostorový model pravostranného mechanismu řazení s detaily



Obrázek 8.2: Detail prostorového modelu se západkovým mechanismem sloužícím k odklápění kabiny

Na detailu mechanismu z obr. 8.2 je zobrazena část mechanismu, která kromě funkce řazení musí v případě odklápění kabiny mechanismus rozpojit pomocí západkového mechanismu, který však při sklopené kabině, během řazení, nemá na funkci mechanismu vliv, proto jej není třeba při kinematické analýze uvažovat.



Obrázek 8.3: Detail řadicí páky a přilehlých součástí

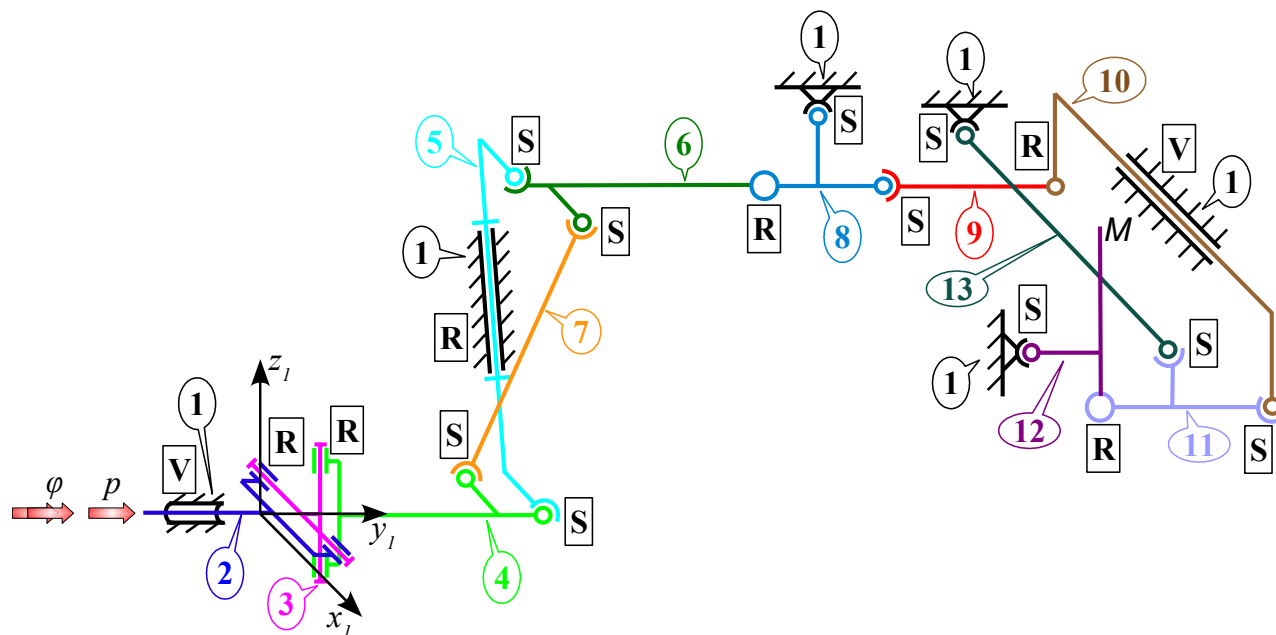
Pokud srovnáme kinematické schéma pravostranného mechanismu řazení z obr. 8.4 s kinematickým schématem levostranného mechanismu viz obr. 7.6, je vidět, že struktury mechanismů jsou zcela odlišné a to z důvodu, že typy nákladních automobilů, pro který jsou mechanismy designovány, jsou také jiné.

Důvodem, proč pravostranný mechanismus, jehož počet členů včetně rámu je 13, je komplikovanější než levostranná varianta, která čítá pouze 8 členů, je motor a jeho příslušenství. Pravostranný mechanismus se musí vyhnout samotnému motoru, přídavnému chlazení, krytu ventilátoru motoru a dalšímu vybavení.

## 8.1 Kinematický rozbor funkce pravostranného mechanismu řazení

V kinematickém schématu uvedeném na obr. 8.4 jsou čísla jednotlivých těles uváděna v bublinách s šipkou směřující k danému členu, vazby jsou značeny písmeny v obdélníku, písmenem S je značen sférický kloub, písmenem R rotační vazba a konečně V označuje vazbu válcovou, která umožňuje posuv po ose a rotaci kolem téže osy.

Při popisu funkce mechanismu bude opět postupováno směrem od převodové skříně k řidiči čili k řadicí páce 12. Převodová skříň je ovládána pomocí dvou stupňů volnosti, posuvu  $p$  po ose  $y_1$  a natočení  $\varphi$  kolem téže osy, tyto dvě souřadnice jsou souřadnicemi mechanismu řazení, protože jednoznačně určují jeho polohu, detailním popisem souřadnic mechanismu se zabývá kap. 7.2. Změna souřadnice  $p$  má za následek pohyb bodu M řadicí páky 12 ve směru jízdy automobilu, tento pohyb bude dále nazýván řazení, naopak změna úhlu  $\varphi$  vyvolá pohyb bodu M ve směru kolmém na směr jízdy automobilu, tento pohyb bude dále nazýván volba.



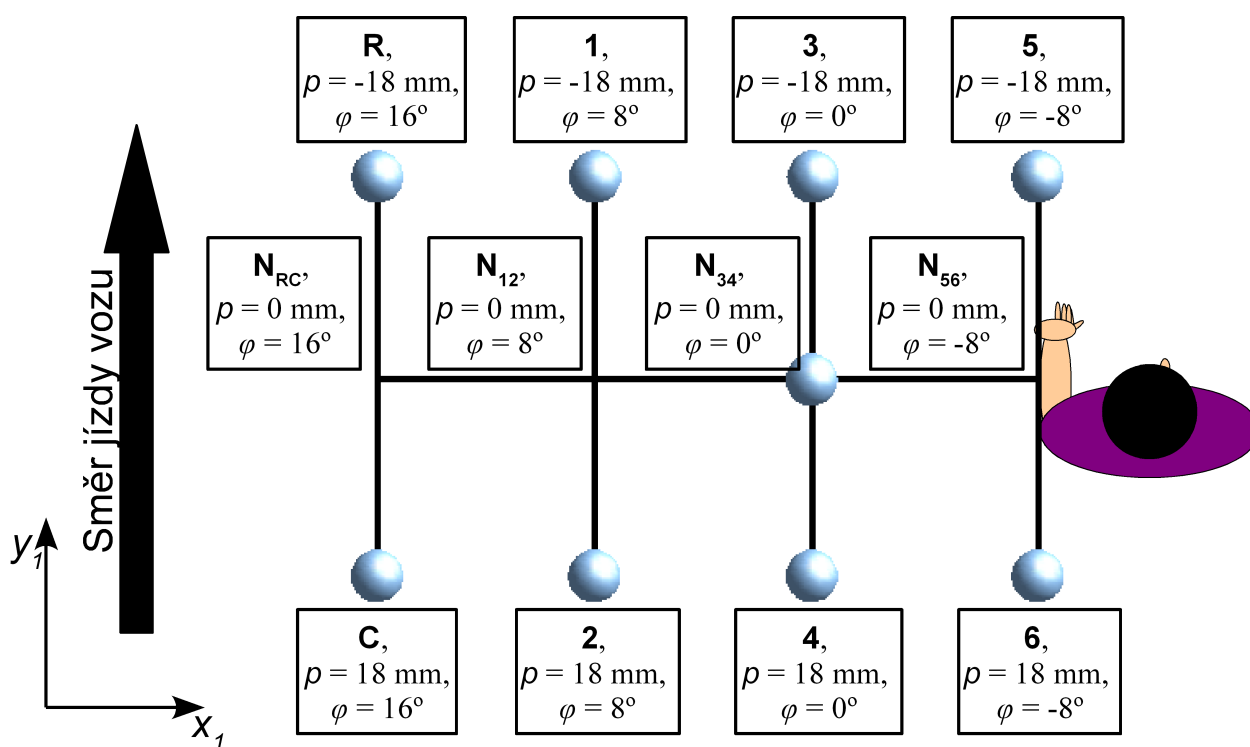
Obrázek 8.4: Kinematické schéma pravostranného mechanismu řazení

Posuv  $p$  a natočení  $\varphi$  jsou aplikovány na člen 2, který je s rámem 1 spojen válcovou vazbou; lze tudíž říci, že posuv  $p$  je posuv souřadného systému členu 2 vůči 1 po ose  $y_1$ , o natočení  $\varphi$  platí analogie k posuvu, pouze se děje kolem osy  $y_1$ , viz obr. 8.7. Na člen 4 jsou přenášeny obě souřadnice mechanismu přes křížový člen 3, neboli Hookův kloub. Pohyb volby je přenášen na těleso 6 z tělesa 4 přes tyč 7, která je vázána k oběma tělesům sférickými klouby, pohyb řazení je přenášen pomocí členu 5, který je taktéž spojen s tělesy 4 a 6 sférickými vazbami a navíc je svázán s rámem 1 rotační vazbou, takže koná rotační pohyb. Člen 6 je dále rotační vazbou spojen s tělesem 8, které koná sférický pohyb, protože je vázáno k rámu 1 sférickým kloubem. Pohyb je ze členu 8 dále přenášen na trámec 10 přes tyč 9, která je vázána ke členu 8 sférickou vazbou a ke členu 10 vazbou rotační. Trámec 10 rotuje kolem osy, po které se současně posouvá, je tudíž spojen s rámem 1 vazbou válcovou. Pomocí sférického kloubu je na trámec 10 vázán člen 11, který se opírá o rám 1 přes tyč 13. Konečně řadicí páka 12 konající sférický pohyb, protože s rámem 1 je svázána sférickým kloubem, je na člen 11 vázána rotační vazbou. Bod  $M$  přísluší rukojeti řadicí páky, jedná se tudíž o bod, jehož trajektorie bude optimalizována.

Globální neboli pevný souřadný systém 1 příslušející rámu 1 je umístěn tak, že jeho počátek je při počáteční poloze (tzn.  $p = 0$  a  $\varphi = 0$ ) na ose vidlice 2, osa  $x_1$  je rovnoběžná s osou válcové vazby, která s rámem 1 spojuje člen 10, poloha osy  $z_1$  je volena tak, aby rovina  $x_1-z_1$  byla rovnoběžná s osou rotační vazby, která váže těleso 5 k rámu 1, a osa  $y_1$  je totožná s osou, po které, respektive kolem které se děje posuv  $p$  respektive natočení  $\varphi$ . Tato volba polohy pevného souřadného systému 1 značně ulehčí sestavování maticových rovnic úlohy polohy.

## 8.2 Rozložení rychlostních stupňů převodové skříně ovládané pravostranným mechanismem a hodnoty souřadnic mechanismu

Pravostranný mechanismus řazení je optimalizován pro ovládání sedmistupňové převodové skříně, která je ovládána pomocí dvou souřadnic mechanismu  $p$  a  $\varphi$ , jak již bylo zmíněno v 7.2. Polohy bodu  $M$  v pevném souřadném systému, které přísluší rychlostním stupňům s odpovídajícími hodnotami souřadnic mechanismu, jsou uvedeny na obr. 8.5. Písmenem R je označen zpětný chod, písmenem C „crawler“ neboli „plížení“, který se používá při vyprošťování nebo v extrémním terénu, písmenem N jsou označeny významné body pohybu zvaného volba a čísla označují rychlostní stupně. Neutrál vozu je umístěn v bodě  $N_{34}$ . Na obr. 8.5 je též schematicky naznačena poloha řidiče a jeho řadicí levé ruky.



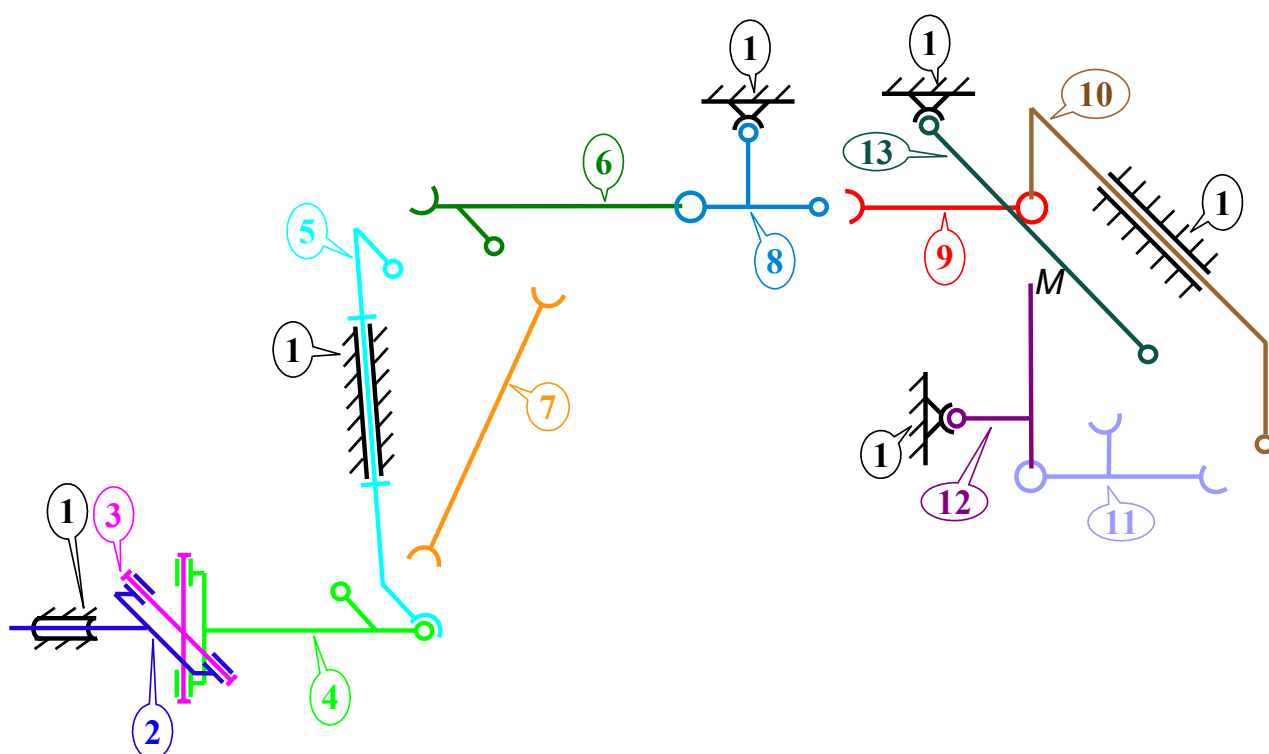
Obrázek 8.5: Rozložení rychlostních stupňů a bodů volby na řadicí páce 12 a jim odpovídající hodnoty souřadnic mechanismu

### 8.3 Stanovení počtu stupňů volnosti a počtu základních kinematických smyček

Počet stupňů volnosti  $i$ , byl určen pomocí vztahu

$$i = 6 \cdot (n - 1) - 3 \cdot S - 4 \cdot V - 5 \cdot R = 6 \cdot (13 - 1) - 3 \cdot 10 - 4 \cdot 2 - 5 \cdot 6 = 4, \quad (8.1)$$

kde velká písmena označují počet vazeb daného typu dle kinematického schématu a písmeno  $n$  označuje počet těles včetně rámu. Z vypočtených čtyřech stupňů volnosti dva připadají souřadnicím mechanismu a zbývající dva stupně volnosti odpovídají nedefinovanému otáčení tyčí 7 a 13 kolem jejich podélné osy.



Obrázek 8.6: Rozložení mechanismu na základní uzavřený řetězec a základní otevřené řetězce

Počet základních smyček byl určen dle kap. 3.2.3. Pro pravostranný mechanismus byl definován základní uzavřený řetězec tvořený členy 1, 2, 3, 4 a 5, a dalších pět otevřených základních řetězců, první otevřený základní řetězec je složen ze členu 7, druhý členy 6 a 8, třetí členy 9 a 10, čtvrtý členy 11 a 12 a pátý členem 13. Rozložení mechanismu na základní uzavřený řetězec a základní otevřené řetězce je naznačeno na schématu z obr. 8.6. Celkový počet základních řetězců je roven šesti, což je současně i počet základních smyček mechanismu. Rozložení základních smyček bylo zvoleno následovně:

- I. Základní smyčka I je složena ze členů 1, 2, 3, 4 a 5.
- II. Základní smyčka II je složena ze členů 4, 5, 6 a 7.

III. Základní smyčka III je složena ze členů 1, 5, 6 a 8.

IV. Základní smyčka IV je složena ze členů 1, 8, 9 a 10.

V. Základní smyčka V je složena ze členů 1, 10, 11 a 12.

VI. Základní smyčka VI je složena ze členů 1, 11, 12 a 13.

Počet neznámých souřadnic vazeb, které budou určovány úlohou polohy, bude

$$r \cdot 6 = 36, \quad (8.2)$$

kde  $r = 6$  je počet základních smyček.

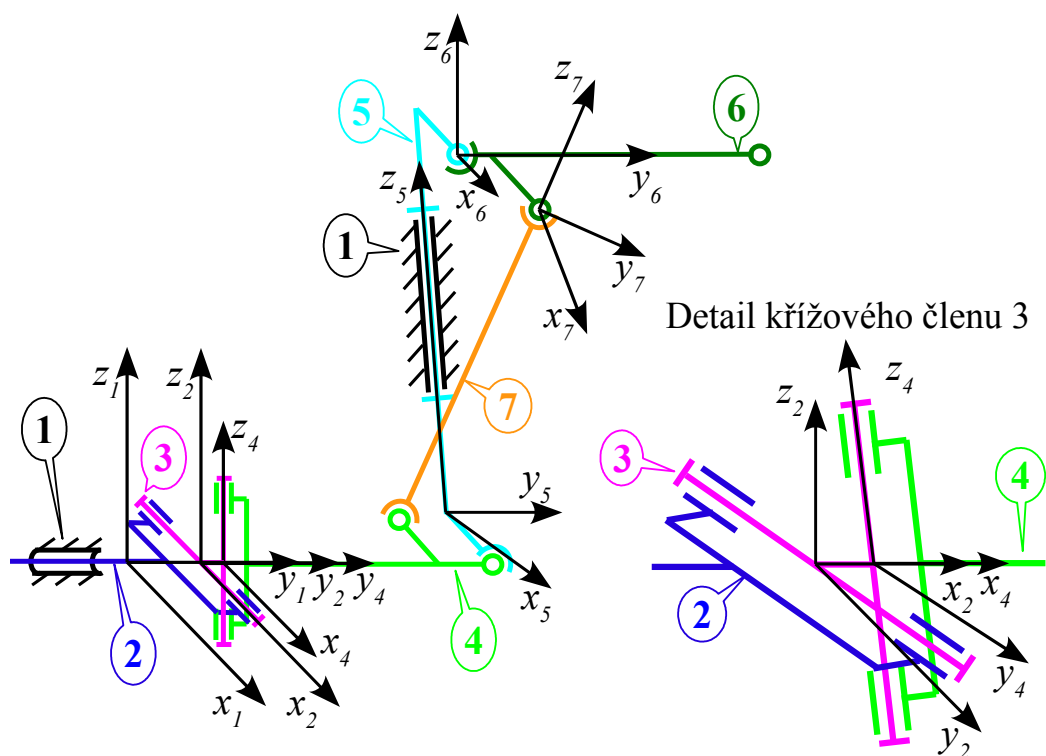
## 8.4 Maticové rovnice úlohy polohy pravostranného mechanismu řazení

Maticové rovnice budou sestavovány pro každou základní smyčku zvlášť, jak bylo uvedeno v kap. 3.2.5. Umístění tělesových souřadných systémů a rozměry mechanismu budou vykresleny z důvodu komplikovanosti mechanismu pro každou smyčku zvlášť. Rozměry pravostranného mechanismu, taktéž z důvodu značné složitosti, budou popisovány hlavně pomocí polohy bodů jak v pevném, tak v tělesových souřadných systémech. Dolní index u písmena, které označuje daný bod, bude mít význam indexu souřadného systému, v němž jsou souřadnice daného bodu zapotřebí; např.  $B_2$  znamená, že budou použity souřadnice bodu  $B$  v souřadném systému 2.

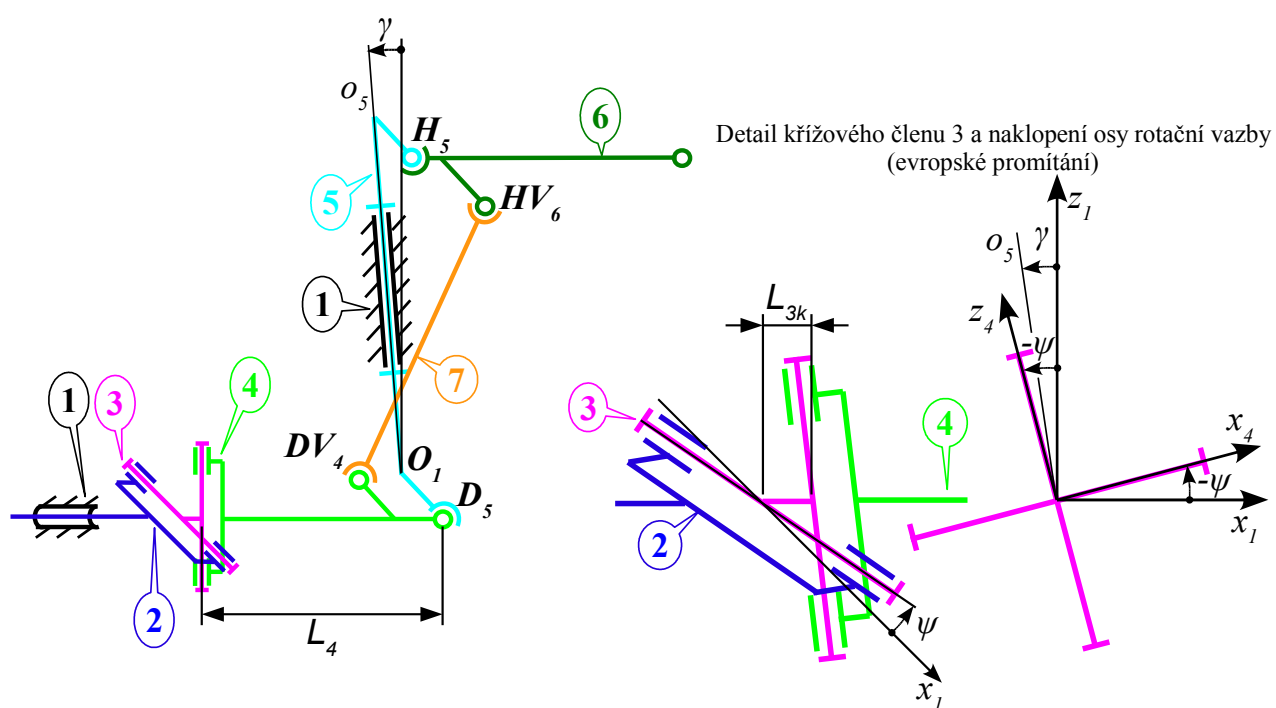
### 8.4.1 Maticová rovnice základní smyčky I

Potřebné rozměry mechanismu jsou uvedeny na obr. 8.7, je vidět, že již v počáteční poloze, tzn.  $p = 0$  a  $\varphi = 0$ , jsou osy křížového členu 3 pootočený vůči pevnému souřadnému systému 1 o úhel  $\psi$  v záporném směru kolem osy  $y_1$ . Dále osa rotační vazby  $o_5$ , která váže těleso 5 k rámu 1, je sklopena o úhel  $\gamma$  kolem osy  $y_1$  ve směru za nákresnu (směr je zvýrazněn i v detailu), osa  $o_5$  leží tudíž v rovině rovnoběžné s rovinou  $z_1-x_1$ . Body  $DV_4$ ,  $D_5$  a  $HV_6$  označují středy sférických kloubů, bod  $H_5$  označuje jak střed sférické vazby, tak i střed souřadného systému členu 6 a bod  $O_1$  střed souřadného systému tělesa 5.





Obrázek 8.7: Umístění tělesových souřadných systémů potřebných při sestavování maticových rovnic základních smyček I a II



Obrázek 8.8: Rozměry potřebné při sestavování maticových rovnic základních smyček I a II

Maticová rovnice základní smyčky I bude mít dle rovnice 3.44 následující tvar

$$\mathbf{T}_{12} \cdot \mathbf{T}_{24} \cdot \mathbf{T}_{45} \cdot \mathbf{T}_{51} = \mathbf{E}, \quad (8.3)$$

kde  $\mathbf{E} \in \mathbb{R}^{4 \times 4}$  je matice identity a  $\mathbf{T}_{ij}$  jsou transformační matice vyjadřující přechod z  $i$ -tého do  $j$ -tého tělesového souřadného systému; tyto matice lze vyjádřit pomocí součinu transformačních matic základních pohybů definovaných v kap. 3.3.2 jako

$$\begin{aligned} \mathbf{T}_{12} &= \mathbf{T}_y(p) \cdot \mathbf{R}_y(\varphi), \\ \mathbf{T}_{24} &= \mathbf{R}_y(-\psi) \cdot \mathbf{R}_x(q_1) \cdot \mathbf{T}_y(L_{3k}) \cdot \mathbf{R}_z(q_2), \\ \mathbf{T}_{45} &= \mathbf{T}_y(L_4) \cdot \mathbf{R}_x(q_3) \cdot \mathbf{R}_y(q_4) \cdot \mathbf{R}_z(q_5) \cdot \mathbf{T}(-\mathbf{D}_5), \\ \mathbf{T}_{51} &= \mathbf{R}_z(q_6) \cdot \mathbf{R}_y(\gamma) \cdot \mathbf{T}(-\mathbf{O}_1). \end{aligned} \quad (8.4)$$

Při vyjadřování transformačních matic byl zaveden vektor  $\mathbf{q} \in \mathbb{R}^{36}$ , jehož prvky jsou neznámé úhly natočení v jednotlivých vazbách a posuv trámce 10. Takže například  $q_1$  je úhel natočení křížového členu 3 vůči členu 2.

#### 8.4.2 Maticová rovnice základní smyčky II

Zavedené souřadné tělesové systémy a rozměry potřebné při sestavování rovnic jsou uvedeny výše na obr. 8.7 a 8.8. Maticovou rovnici základní smyčky II lze psát ve tvaru

$$\mathbf{T}_{45} \cdot \mathbf{T}_{56} \cdot \mathbf{T}_{67} \cdot \mathbf{T}_{74} = \mathbf{E}, \quad (8.5)$$

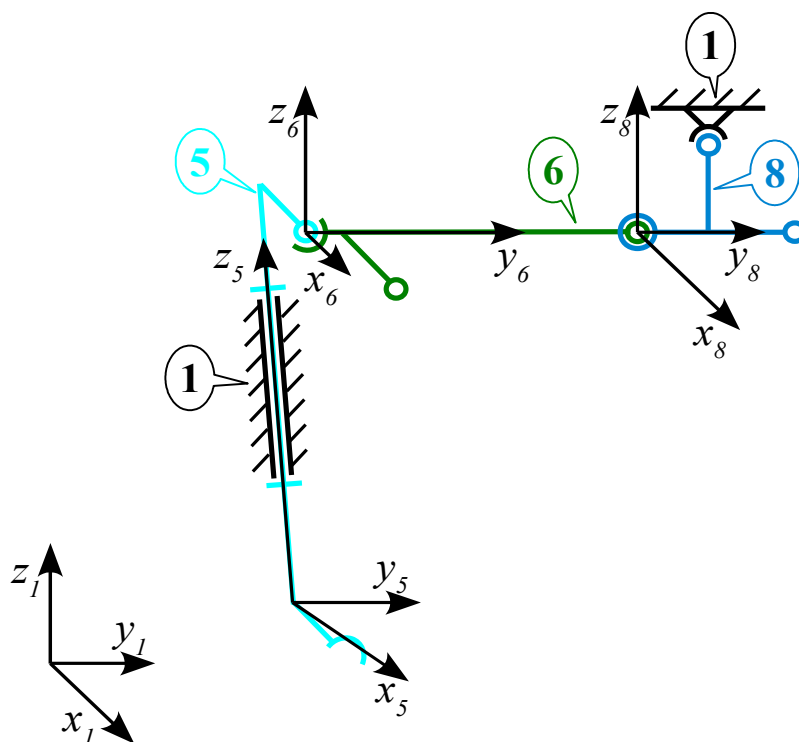
transformační matice lze opět rozepsat pomocí základních pohybů

$$\begin{aligned} \mathbf{T}_{56} &= \mathbf{T}(\mathbf{H}_5) \cdot \mathbf{R}_x(q_7) \cdot \mathbf{R}_y(q_8) \cdot \mathbf{R}_z(q_9), \\ \mathbf{T}_{67} &= \mathbf{T}(\mathbf{H}\mathbf{V}_6) \cdot \mathbf{R}_x(q_{10}) \cdot \mathbf{R}_y(q_{11}) \cdot \mathbf{R}_z(q_{12}), \\ \mathbf{T}_{74} &= \mathbf{T}_z(L_7) \cdot \mathbf{R}_x(q_{13}) \cdot \mathbf{R}_y(q_{14}) \cdot \mathbf{T}(-\mathbf{D}\mathbf{V}_4), \end{aligned} \quad (8.6)$$

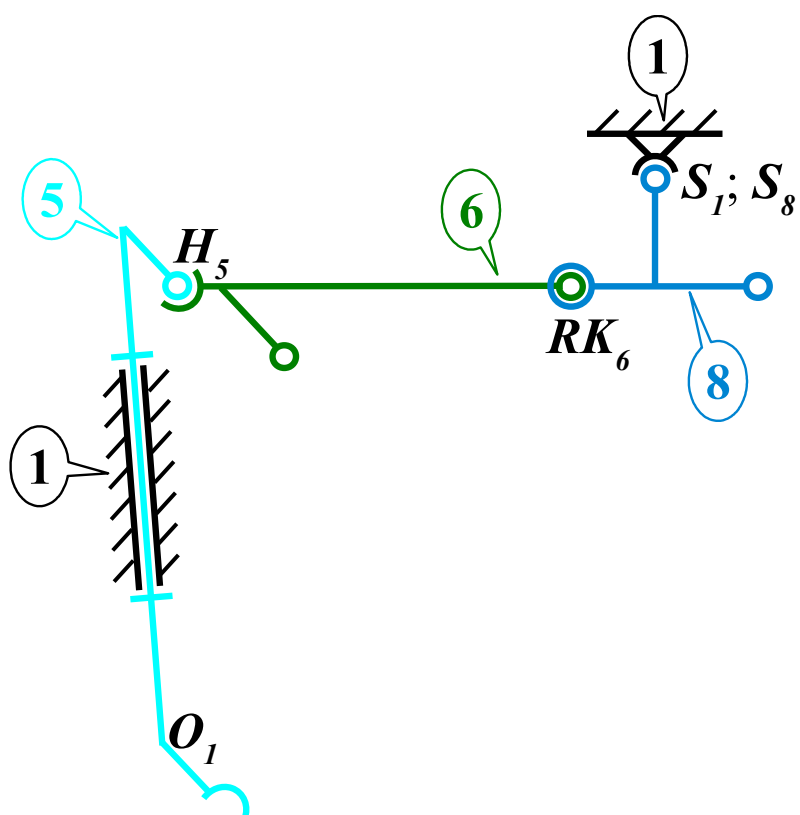
matice  $\mathbf{T}_{45}$  byla již rozepsána výše v 8.4. U transformační matice  $\mathbf{T}_{74}$ , která vyjadřuje transformaci z tělesového prostoru tyče 7 do prostoru členu 4 (tělesa jsou vázána sférickým kloubem, který má tři stupně volnosti respektive je popsán třemi natočeními), je vynechána rotace kolem osy  $z$ , čímž je eliminováno nedefinované natáčení tyče 7 kolem její podélné osy.

#### 8.4.3 Maticová rovnice základní smyčky III

Zavedené tělesové souřadné systémy potřebné při sestavování maticové rovnice základní smyčky III jsou uvedeny na obr. 8.9 a na následujícím obr. 8.10 jsou vyznačeny potřebné body. Body  $\mathbf{O}_7$  a  $\mathbf{H}_5$  byly již popsány výše, bod  $\mathbf{RK}_6$  leží v počátku souřadného systému 8 a na ose rotační vazby, dále bude pro výpočet zapotřebí znát polohu středu sférického kloubu  $\mathbf{S}_7$  a  $\mathbf{S}_8$  v obou souřadných systémech 1 a 8.



Obrázek 8.9: Zavedení tělesových souřadných systémů potřebných při sestavování maticových rovnic základní smyčky III



Obrázek 8.10: Rozměry potřebné při sestavování maticových rovnic základní smyčky III

Nyní, když jsou zavedeny tělesové souřadné systémy a rozměry smyčky, můžeme psát maticovou rovnici základní smyčky III

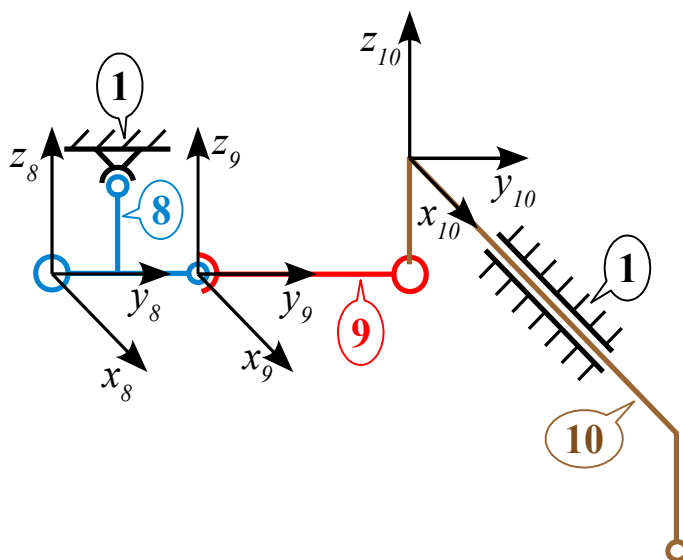
$$T_{15} \cdot T_{56} \cdot T_{68} \cdot T_{81} = E, \quad (8.7)$$

transformační matice lze rozepsat jako

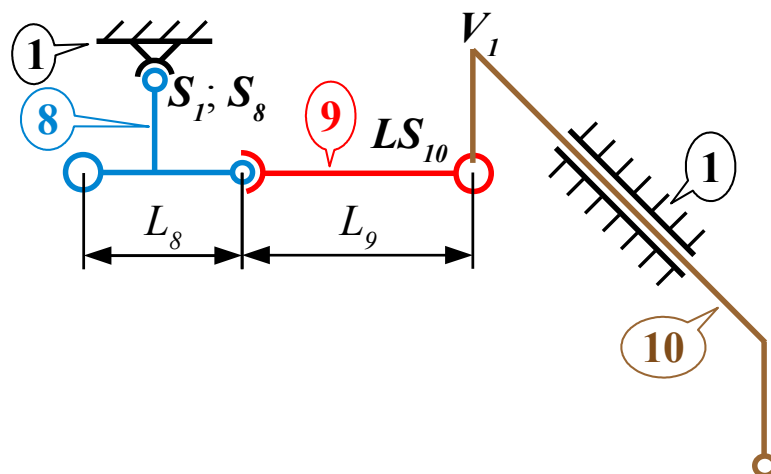
$$\begin{aligned} T_{15} &= T(O_1) \cdot R_y(-\gamma) \cdot R_z(-q_6), \\ T_{68} &= T(RK_6) \cdot R_x(q_{15}), \\ T_{81} &= T(S_8) \cdot R_x(q_{16}) \cdot R_y(q_{17}) \cdot R_z(q_{18}) \cdot T(-S_1), \end{aligned} \quad (8.8)$$

matice  $T_{56}$  byla již uvedena výše v 8.6.

#### 8.4.4 Maticová rovnice základní smyčky IV



Obrázek 8.11: Zavedení tělesových souřadných systémů potřebných při sestavování maticových rovnic základní smyčky IV



Obrázek 8.12: Rozměry potřebné při sestavování maticových rovnic základní smyčky IV

Zavedené tělesové souřadné systémy základní smyčky IV jsou uvedeny na obr. 8.11 a na následujícím obr. 8.12 jsou uvedeny potřebné rozměry, bod  $S$  byl již popsán výše, bod  $LS_{10}$  leží na ose rotační vazby a na ose  $y_9$  a bod  $V_I$  leží v počátku souřadného systému trámce 10.

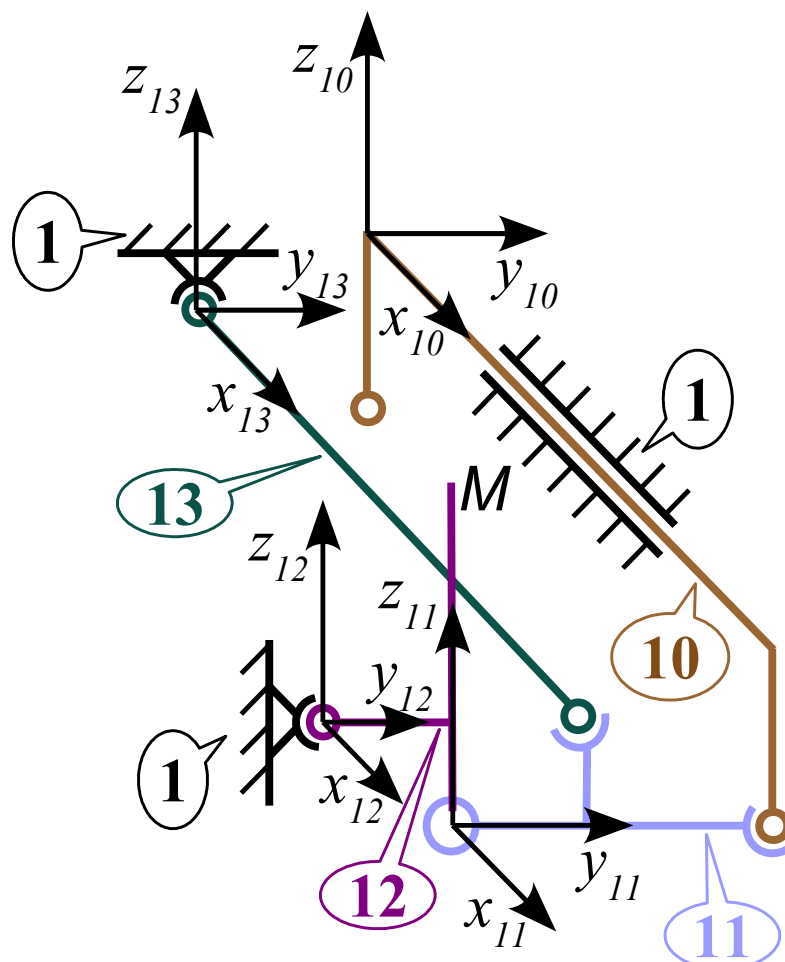
Maticová rovnice základní smyčky IV má tvar

$$T_{18} \cdot T_{89} \cdot T_{9,10} \cdot T_{10,I} = E, \quad (8.9)$$

transformační matice je třeba opět rozvést do tvaru

$$\begin{aligned} T_{18} &= T(S_I) \cdot R_z(-q_{18}) \cdot R_y(-q_{17}) \cdot R_x(-q_{16}) \cdot T(-S_8), \\ T_{89} &= T_y(L_8) \cdot R_x(q_{19}) \cdot R_y(q_{20}) \cdot R_z(q_{21}), \\ T_{9,10} &= T_y(L_9) \cdot R_x(q_{22}) \cdot T(-LS_{10}), \\ T_{10,I} &= T_x(q_{23}) \cdot R_x(q_{24}) \cdot T(-V_I). \end{aligned} \quad (8.10)$$

#### 8.4.5 Maticová rovnice základní smyčky V



Obrázek 8.13: Zavedení tělesových souřadných systémů potřebných při sestavování maticových rovnic základních smyček V a VI

Zavedení polohy tělesových souřadných systémů a rozměry vztahující se k základní

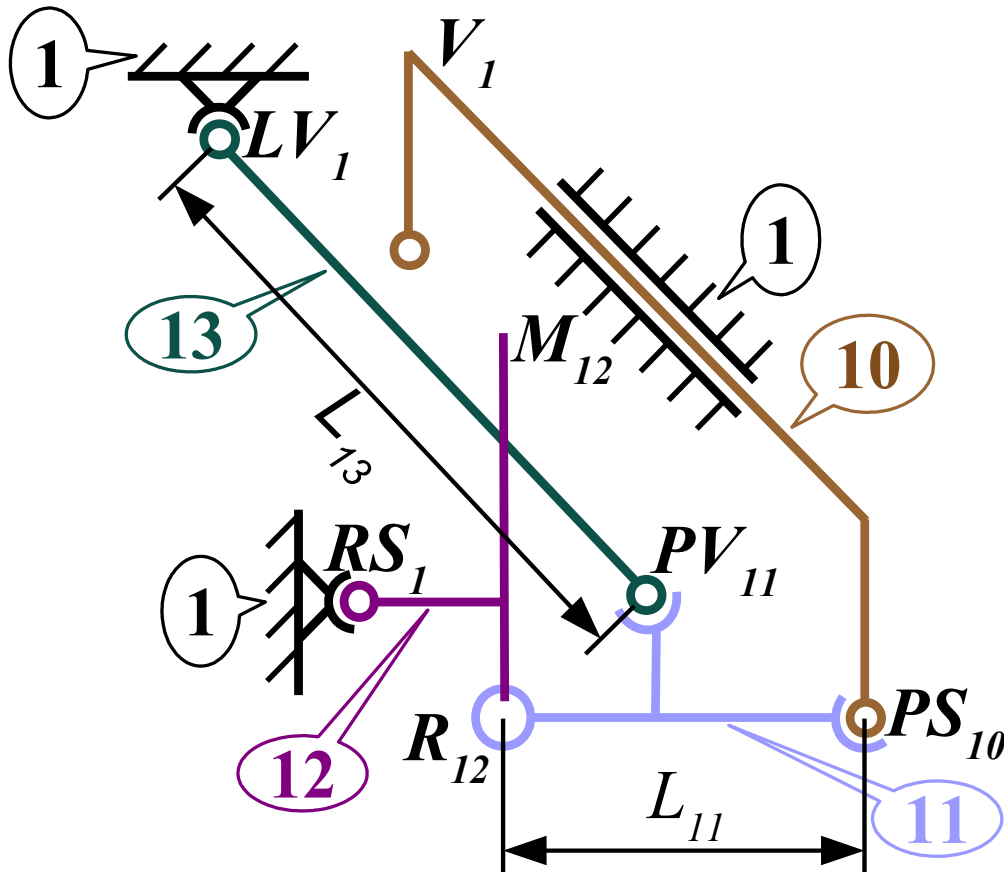
smyčky V jsou uvedeny na následujících dvou obr. 8.13 a 8.14. Bod  $V_1$  byl již popsán výše, body  $PS_{10}$ ,  $PV_{11}$ ,  $LV_1$  a  $RS_1$  odpovídají středům sférických kloubů, bod  $R_{12}$  leží na ose rotační vazby a bod  $M_{12}$  označuje bod rukojeti řadící páky 12, jehož dráha v pevném souřadném systému je optimalizována.

Nyní lze psát maticovou rovnici základní smyčky V

$$T_{1,10} \cdot T_{10,11} \cdot T_{11,12} \cdot T_{12,1} = E, \quad (8.11)$$

jednotlivé matice je třeba rozepsat jako součin matic základních pohybů

$$\begin{aligned} T_{10,1} &= T(V_1) \cdot R_x(-q_{24}) \cdot T_x(-q_{23}), \\ T_{10,11} &= T(PS_{10}) \cdot R_x(q_{25}) \cdot R_y(q_{26}) \cdot R_z(q_{27}) \cdot T_y(-L_{11}), \\ T_{11,12} &= R_x(q_{28}) \cdot T(-R_{12}), \\ T_{12,1} &= R_x(q_{29}) \cdot R_y(q_{30}) \cdot R_z(q_{31}) \cdot T(-RS_1). \end{aligned} \quad (8.12)$$



Obrázek 8.14: Rozměry potřebné při sestavování maticových rovnic základních smyček V a VI

#### 8.4.6 Maticová rovnice základní smyčky VI

Umístění tělesových souřadných systémů i rozměry, které jsou využity při formulaci maticové rovnice, jsou uvedeny na předcházejících obr. 8.13 a 8.14. Lze tudíž přímo psát maticovou

rovnici základní smyčky VI ve tvaru

$$\mathbf{T}_{1,12} \cdot \mathbf{T}_{12,11} \cdot \mathbf{T}_{11,13} \cdot \mathbf{T}_{13,1} = \mathbf{E}, \quad (8.13)$$

kde jednotlivé matice je třeba rozvést na tvar

$$\begin{aligned} \mathbf{T}_{1,12} &= \mathbf{T}(\mathbf{RS}_1) \cdot \mathbf{R}_z(-q_{31}) \cdot \mathbf{R}_y(-q_{30}) \cdot \mathbf{R}_x(-q_{29}), \\ \mathbf{T}_{12,11} &= \mathbf{T}(\mathbf{R}_{12}) \cdot \mathbf{R}_x(-q_{28}), \\ \mathbf{T}_{11,13} &= \mathbf{T}(\mathbf{PV}_{11}) \cdot \mathbf{R}_x(q_{32}) \cdot \mathbf{R}_y(q_{33}) \cdot \mathbf{R}_z(q_{34}) \cdot \mathbf{T}_x(-L_{13}), \\ \mathbf{T}_{13,1} &= \mathbf{R}_y(q_{35}) \cdot \mathbf{R}_z(q_{36}) \cdot \mathbf{T}(-\mathbf{LV}_1), \end{aligned} \quad (8.14)$$

u transformační matice  $\mathbf{T}_{13,1}$ , která vyjadřuje transformaci z tělesového prostoru tyče 13 do prostoru rámu 1 (tělesa jsou vázána sférickým kloubem, který má tři stupně volnosti respektive je popsán třemi natočeními), je vynechána rotace kolem osy  $x$ , čímž je eliminováno nedefinované natáčení tyče 13 kolem její podélné osy.

#### 8.4.7 Shrnutí úlohy polohy

Nyní máme k dispozici šest maticových rovnic 8.3, 8.5, 8.7, 8.9, 8.11 a 8.13, z nichž z každé lze použít pouze šest skalárních rovnic dle vzoru 3.45, takže bylo tímto postupem získáno třicet šest transcendentních rovnic, v nichž neznámými je třicet šest prvků vektoru  $\mathbf{q} \in \mathbb{R}^{36}$ , což jsou hodnoty úhlů natočení jednotlivých vazeb a posuv ve válcové vazbě.

#### 8.4.8 Postup výpočtu trajektorie bodu M

Polohu bodu M rukojeti řadicí páky 12 v globálních souřadnicích 1 lze vyjádřit pomocí rozšířeného průvodiče bodu M jako

$$\mathbf{r}_{1M} = \mathbf{T}_{1,12} \cdot \mathbf{r}_{12M}, \quad (8.15)$$

kde  $\mathbf{T}_{1,12}$  je rozepsána v 8.14 a prvky rozšířeného průvodiče  $\mathbf{r}_{12M}$  bodu  $\mathbf{M}_{12}$  v tělesovém souřadném systému 12 řadicí páky 12 jsou známy, struktura vektoru je následující

$$\mathbf{r}_{12M} = \begin{bmatrix} x_{12M} \\ y_{12M} \\ z_{12M} \\ 1 \end{bmatrix}. \quad (8.16)$$

Postup výpočtu trajektorie bodu  $\mathbf{M}$  při řazení rychlostních stupňů spočívá ve výpočtu úlohy polohy (dle rovnic získaných ze základních smyček) pro ty souřadnice mechanismu, které odpovídají rychlostním stupňům a bodům volby, viz obr. 8.5. Jakmile je vypočtena úloha polohy, lze stanovit polohu bodu  $\mathbf{M}$  v pevném souřadném systému 1 dle vztahu 8.15. Tyto body postačují ke stanovení hodnoty cílové funkce, pokud bychom chtěli získat hladší trajektorie, je zapotřebí stanovit

úlohu polohy a polohu bodu  $M$  ve více bodech.

Algoritmus výpočtu úlohy polohy pravostranného mechanismu řazení a hodnoty cílové funkce (viz dále) napsaný v programovacím jazyku používaném v softwaru MATLAB je přiložen v 11.11.

## 8.5 Cílová funkce

Volba funkčního předpisu cílové funkce je podrobněji rozpracována v kap. 7.7, takže na tomto místě je uvedeno pouze nezbytné shrnutí a úprava vztahů, protože globální souřadný systém 1 pravostranného mechanismu řazení byl zvolen s jinou orientací souřadných os než pro mechanismus levostranný. Polohy i označení významných bodů drah řazení a orientace globálního souřadného systému 1 pro pravostranný mechanismus jsou uvedeny na obr. 8.5. Vůči kapitole zabývající se řazením levostranným se bude cílová funkce lišit pouze tím, že počet drah řazení, jejichž tvar je optimalizován, je tentokrát roven čtyřem. Zavedme symbolické indexy  $i$  a  $j$ , které označují jednotlivé body drah řazení, a index  $k$ , který značí jednotlivé dráhy řazení.

- Pro dráhu řazení z bodu C do bodu R položme  $i = C, j = R$  a  $k = CR$ .
- Pro dráhu řazení z bodu 1 do bodu 2 položme  $i = 1, j = 2$  a  $k = 12$ .
- Pro dráhu řazení z bodu 3 do bodu 4 položme  $i = 3, j = 4$  a  $k = 34$ .
- Pro dráhu řazení z bodu 5 do bodu 6 položme  $i = 5, j = 6$  a  $k = 56$ .

Nyní, po zavedení symbolických indexů, lze vyjádřit cílovou funkci jako součet funkcí  $G_k$  jednotlivých drah řazení

$$F = \sum_k G_k \quad (8.17)$$

kde  $k$  značí výše uvedené symbolické indexy jednotlivých drah řazení a nabývá hodnot  $k \in CR; 12; 34; 56$ .

Při vyjádření funkcí  $G_k$  bude třeba znát délku drah  $L_d$ , která je vyjádřena (za předpokladu, že délka jednotlivých drah se výrazně neliší) jako

$$L_d = y_R - y_C. \quad (8.18)$$



### 8.5.1 Funkce jednotlivých drah řazení

Sklon  $s_k$   $k$  – té dráhy je vyjádřen pomocí rozdílu  $x$ -ových souřadnic bodů  $i$  a  $j$

$$s_k = x_i - x_j. \quad (8.19)$$

Křivost  $d_k$   $k$  – té dráhy je vyjádřena jako vzdálenost přímky procházející body  $i$  a  $j$  od bodu  $N_k$ . Necht' rovnice přímky procházející body  $i$  a  $j$  má následující tvar

$$a_k \cdot x + b_k \cdot y + c_k = 0, \quad (8.20)$$

kde  $a_k = y_i - y_j$ ,  $b_k = x_j - x_i$  a  $c_k = x_i \cdot (y_j - y_i) - y_i \cdot (x_j - x_i)$ ,

pak vzdálenost (křivost)  $d_k$  bude

$$d_k = \frac{a_k \cdot x_k + b_k \cdot y_k + c_k}{\sqrt{a_k^2 + b_k^2}}, \quad (8.21)$$

kde  $x_k$  a  $y_k$  označují souřadnice bodu  $N_k$ .

Hodnotu funkce  $G_k$  dráhy  $k$  můžeme nyní vyjádřit jako

$$G_k = \left( \frac{w_s \cdot s_k}{L_d} \right)^2 + \left( \frac{w_d \cdot d_k}{L_d} \right)^2, \quad (8.22)$$

kde skaláry nabývající hodnot  $w_s, w_d \in \langle 0; 1 \rangle$  jsou váhové faktory; je vhodné volit je tak, aby platilo

$$w_s + w_d = 1. \quad (8.23)$$

Těmito faktory lze zesilovat či zeslabovat dopad jedné za dvou stanovených vlastností, například položíme-li  $w_s = 1$  a  $w_d = 0$ , bude optimalizován pouze sklon drah řazení. Pokud nebude uvedeno jinak, budou váhové faktory rovny  $w_s = w_d = 0,5$ .

## 8.6 Rozměry pravostranného mechanismu řazení

Všechny rozměry mechanismu, které byly použity při sestavování úlohy polohy pomocí maticových rovnic základních smyček, jsou uvedeny v tab. 8.1; význam rozměrů vyplývá z obrázků uvedených u jednotlivých základních smyček výše. Rozměry mechanismu byly měřeny na prostorovém počítačovém modelu dodaném výrobcem.

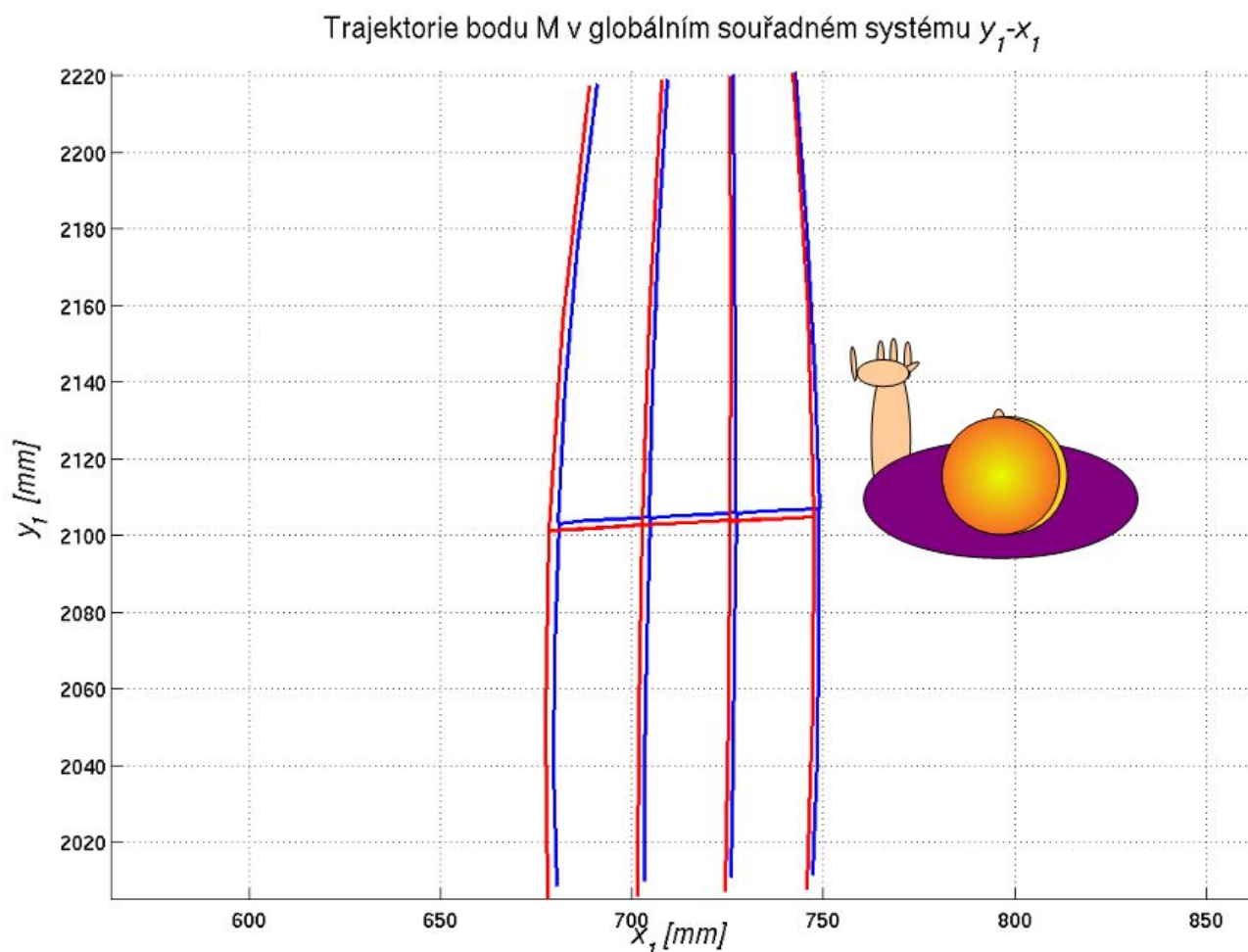
Tabulka 8.1: Rozměry pravostranného mechanismu řazení

Úhlové rozměry		Souřadnice bodů	
Označení úhlu	Hodnota úhlu [°]	Označení bodu	Souřadnice bodu [mm] <sup>19</sup>
$\psi$	6	$DV_4$	[-64; 842,5; -1,5]
$\gamma$	7	$D_5$	[61,0; 0,0; 0,0]
Délkové rozměry		$O_1$	[-34,0; 936,0; 41,5]
Označení délky	Hodnota [mm]	$H_5$	[61,0; 0,0; 711,5];
$L_{3k}$	16,0	$HV_6$	[67,5; 13,5; 1,5]
$L_4$	922,0	$RK_6$	[2,0; 1089,0; 0,0]
$L_7$	726,0	$S_8$	[-0,5; 44,5; 59,0]
$L_8$	50,0	$S_1$	[-81,5; 2069,0; 768,0]
$L_9$	265,0	$LS_{10}$	[0,0; 0,0; -100,0]
$L_{11}$	265,0	$V_1$	[-81,0; 2339,5; 809,0]
$L_{13}$	758,0	$PS_{10}$	[789,0; 0,0; -99,0]
		$R_{12}$	[0,0; 5,5; -59,0]
		$RS_1$	[708,0; 2069,0; 768,5]
		$PV_{11}$	[-17,0; 173,5; 97,5]
		$LV_1$	[-67,5; 2236,5; 808,0]
		$M_{12}$	[17,0; 35,0; 348,0]

## 8.7 Dráhy řazení a srovnání matematického modelu mechanismu s hodnotami získanými z programu Pro/ENGINEER

Na obr. 8.15 jsou zobrazeny trajektorie bodu M rukojeti řadicí páky 12 pravostranného mechanismu řazení, modrá trajektorie byla vypočtena pomocí matematického modelu reprezentovaného maticovými rovnicemi uvedenými výše za použití výše uvedených rozměrů, naopak červená trajektorie byla získána pomocí modulu „Motion“ implementovaném v softwaru Pro/ENGINEER. Pokud trajektorie srovnáme, lze prohlásit, že tvar obou drah řazení je téměř stejný, pouze jsou vůči sobě posunuty o vzdálenost cca 2 mm. Drobná odchylka byla způsobena s největší pravděpodobností během měření a také zaokrouhlením - naměřené hodnoty byly totiž zaokrouhleny s přesností 0,5 mm. Vzájemný posuv drah o 2 mm lze posoudit jako nepodstatný, protože optimalizován je tvar drah a ten je zachován.

<sup>19</sup> Souřadnice bodu je vyjádřena v souřadném systému, jehož index odpovídá dolnímu indexu u označení bodu.



Obrázek 8.15: Dráhy řazení pravostranného mechanismu řazení

Hodnota cílové funkce pro pravostranný mechanismus řazení s původními rozměry je

$$F = 11 \cdot 10^{-4}. \quad (8.24)$$

Po konzultaci s výrobcem byl shledán vypočtený tvar drah řazení uspokojivým, ale bohužel tvar drah na vyrobených nákladních automobilech neodpovídal tvaru vypočtenému a navíc se lišil pro každý vyrobený automobil. Tento jev lze vysvětlit pouze přítomností výrobních a montážních imperfekcí, které způsobí, že pro každý vyrobený kus jsou dráhy řazení jiné.

Cesty, jak vyřešit vzniklý problém jsou dvě:

Prvním a také méně radikálním řešením je zvýšit přesnost výroby a zvláště montáže; bylo by tudíž zapotřebí identifikovat rozměry, na jejichž změnu je mechanismus citlivý. Za tímto účelem byla provedena citlivostní analýza, viz následující kap. 8.8.

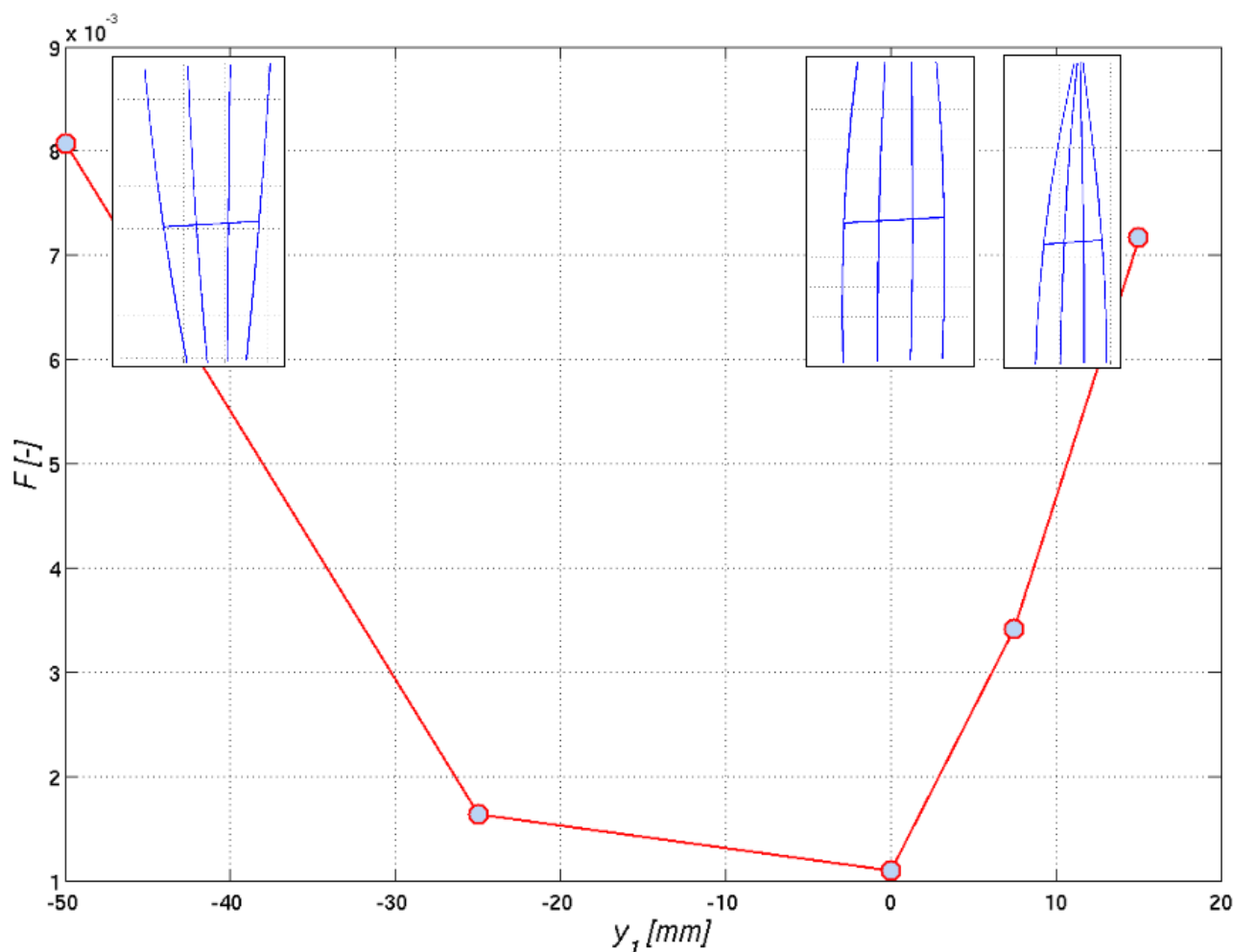
Druhé řešení, jímž se zabývá kap. 8.9, je změna struktury mechanismu tak, aby stabilita variantního návrhu vzhledem k vzniklým výrobním a montážním odchylkám byla větší.

## 8.8 Citlivostní analýza

Citlivostní analýza bude vykreslována do grafů, kde na vertikální ose bude vynesena hodnota cílové funkce a na horizontální ose hodnoty parametru, jehož citlivost je zkoumán; dále bude graf doplněn o obrázky drah řazení pro krajní hodnoty a pro původní hodnotu parametru. Vzhledem k tomu, že se jedná o rozsáhlou část, budou v hlavní části práce uvedeny pouze citlivostní grafy kritických rozměrů. Ostatní citlivostní analýzy jsou uvedeny v příloze 11.15.

### 8.8.1 Změna polohy kabiny ve směru $y_1$

Citlivost mechanismu na změnu polohy kabiny ve směru osy  $y_1$  je uvedena v grafu a tabulce, které jsou uvedeny níže. Je zajímavé, že zvláště posuv kabiny v kladném směru osy  $y_1$  značně bortí tvar drah řazení, a to již při posuvu o 15 mm, v případě opačného posunu kabiny není zborcení tak markantní.



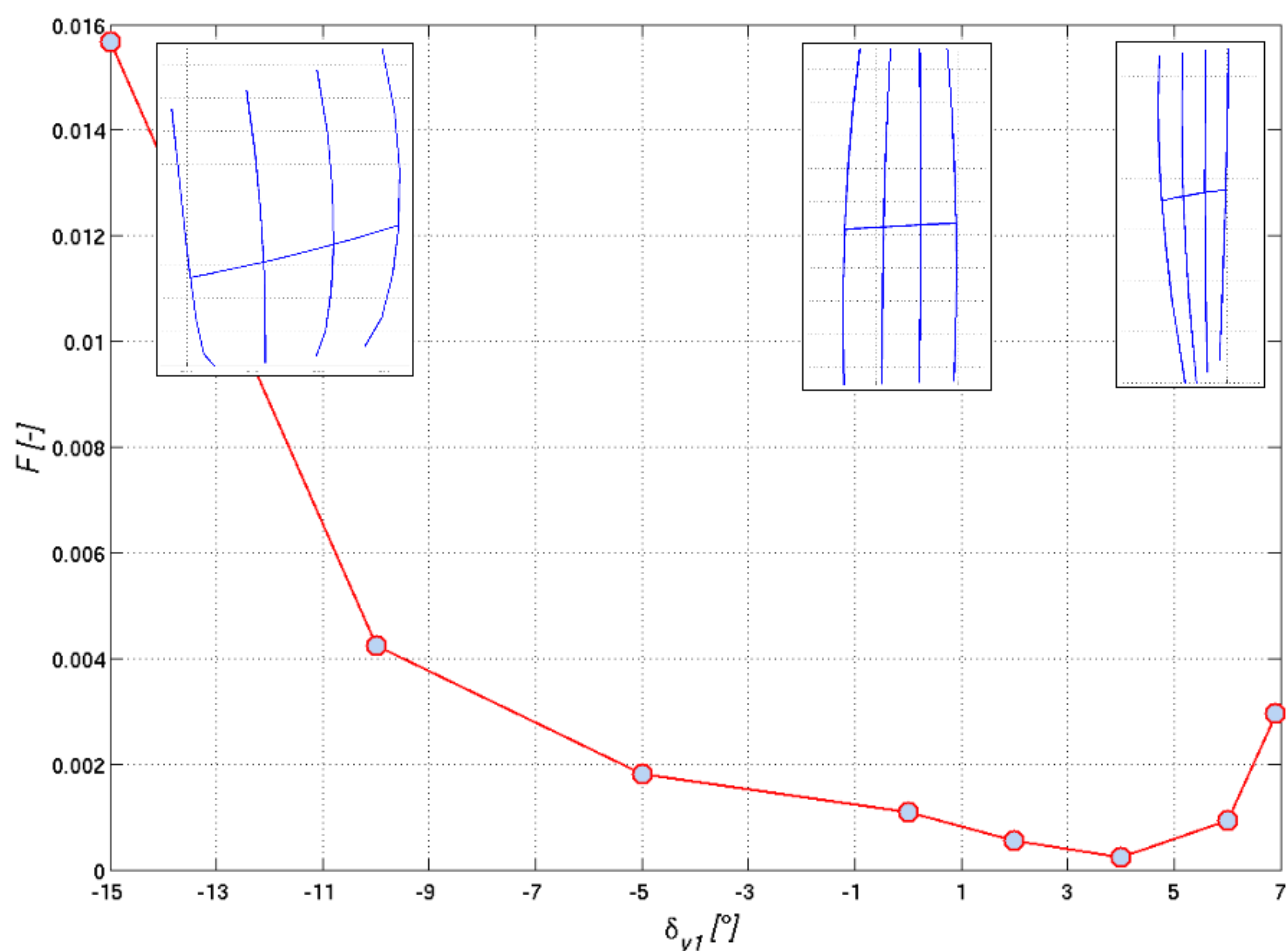
Obrázek 8.16: Citlivost na změnu polohy kabiny ve směru  $y_1$

Tabulka 8.2: Hodnoty cílové funkce v závislosti na posuvu kabiny ve směru  $y_1$ 

$y_1$ [mm]	-50	-25	0	7,5	15
$F \cdot 10^4$ [-]	81	16	11	34	72

### 8.8.2 Naklopení osy válcové vazby

#### Naklopení osy válcové vazby spojující trámec 10 s rámem 1 kolem $y_1$

Obrázek 8.17: Citlivost na naklopení osy válcové vazby spojující trámec 10 s rámem 1 kolem osy  $y_1$ 

Další imperfekcí, která může vzniknout při výrobě a následné montáži, je naklopení osy válcové vazby, která váže trámec 10 k rámu 1. Osa rotace byla při zkoumání citlivosti natáčena kolem osy  $y_1$ , která byla posunuta do bodu  $V_1$  o úhel  $\delta_{y1}$ , jehož kladná orientace je v souladu s pravidlem pravé ruky vzhledem k ose  $y_1$ . Výsledky analýzy, které jsou shrnuty v obr. 8.17 a tab. 8.3, ukazují zajímavé vlastnosti mechanismu, a to, že naklápění osy v záporném smyslu výrazně neovlivní tvar drah řazení; naopak v kladném smyslu dojde při naklopení osy nejdříve k zlepšení hodnoty cílové funkce a následovně ke kritickému zlomu a tvar drah se začne prudce zhoršovat. Lze vyvodit závěr, že při montáži a výrobě by měl být dodržen tento úhel mírně kladný (tolerance závisí

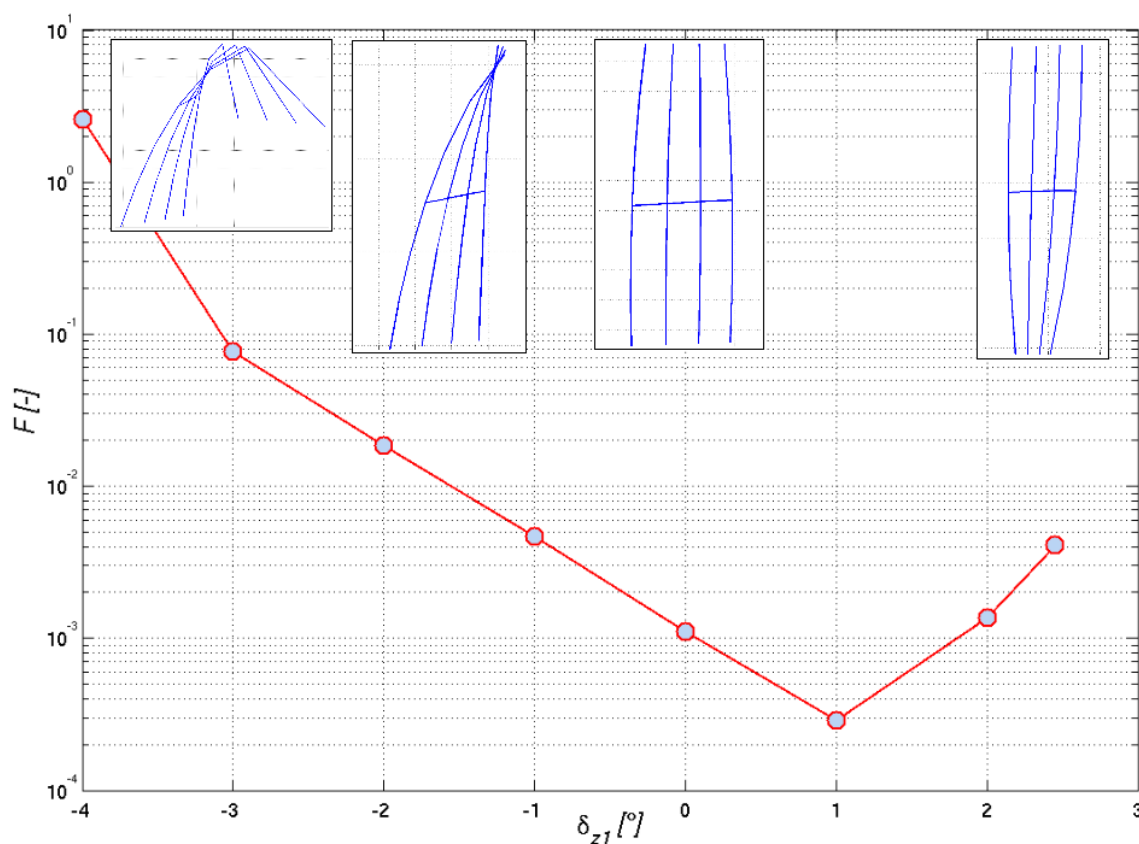
na možnostech výrobce), ale neměl by přesahovat  $\delta_{y1} = 4^\circ$ , protože chování mechanismu za touto hranicí se zdá být již silně nestabilní.

Tabulka 8.3: Hodnoty cílové funkce v závislosti na úhlu naklonění osy válcové vazby spojující trámec 10 s rámem 1 kolem osy  $y_1$

$\delta_{y1} [^\circ]$	-15	-10	-5	0	2	4	6	6,9
$F \cdot 10^4 [-]$	157	42	18	11	6	2	9	30

### Naklonění osy válcové vazby spojující trámec 10 s rámem 1 kolem $z_1$

Tato analýza je analogická k předchozí, tentokrát však bude zkoumán vliv naklonění osy válcové vazby kolem osy  $z_1$  o úhel  $\delta_{z1}$ . Z níže uvedeného obr. 8.18 (pro lepší přehlednost bylo použito pro vertikální osu logaritmické měřítko) je vidět, že vliv změny tohoto parametru pravostranného mechanismu řazení na průběh drah řazení je kritický. Natočení osy v záporném smyslu již o úhel  $\delta_{z1} = -2^\circ$  způsobí kompletní zborcení tvaru drah řazení; naopak při natočení osy v kladném smyslu dochází zpočátku k zlepšení, avšak při natočení osy o větší úhel než  $\delta_{z1} = 2,45^\circ$  již úloha polohy nekonvergovala, proto platí podobné doporučení jako v minulé podkapitole, avšak vliv změny úhlu  $\delta_{z1}$  se ukazuje o mnoho větší než v případě úhlu  $\delta_{y1}$ .



Obrázek 8.18: Citlivost na naklonění osy válcové vazby spojující trámec 10 s rámem 1 kolem osy  $z_1$ ; dráhy řazení jsou seřazeny zleva s hodnotami  $\delta_{z1} = -4^\circ; -3^\circ; 0^\circ; 2,45^\circ$

Tabulka 8.4: Hodnoty cílové funkce v závislosti na úhlu naklonění osy válcové vazby spojující trámec 10 s rámem 1 kolem osy  $z_1$

$\delta_{z1} [^\circ]$	-4	-3	-2	-1	0	1	2	2,45
$F \cdot 10^4 [-]$	2600	770	185	47	11	3	14	41

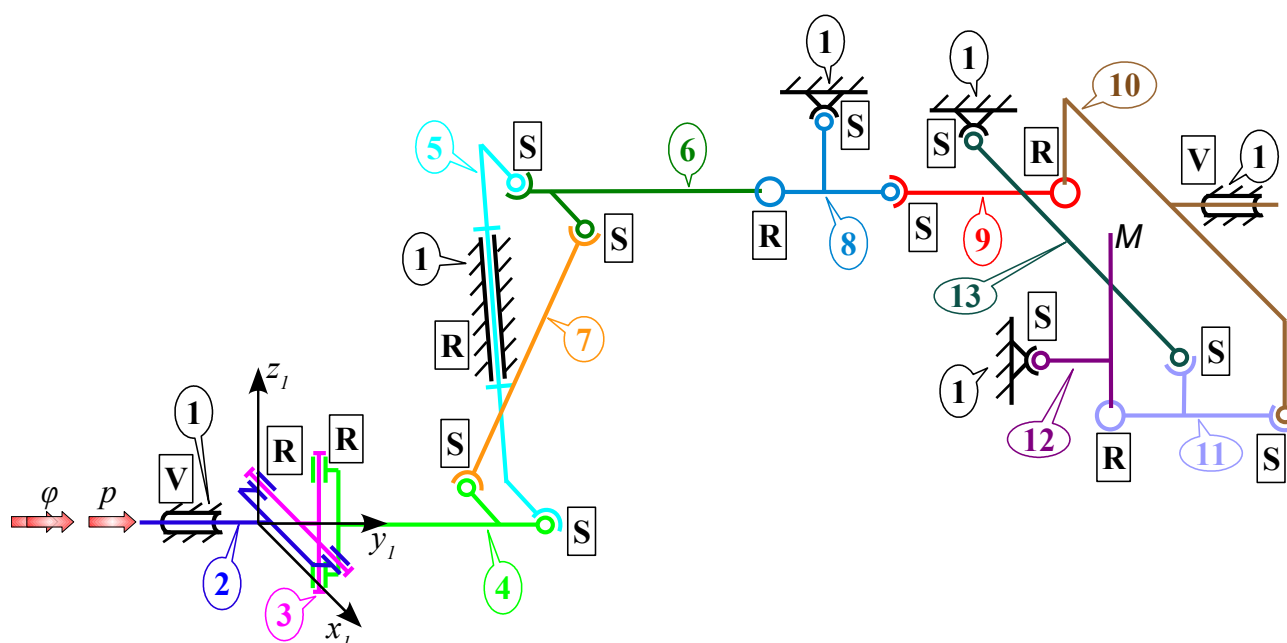
### 8.8.3 Shrnutí výsledků citlivostní analýzy

Provedená analýza odhalila kritický rozměr, jímž je naklonění osy válcové vazby kolem osy  $z_1$ , která váže trámec 10 s rámem 1, viz obr. 8.18. Změna tohoto úhlu již o  $-2^\circ$  zapříčiní zborcení drah. Dalšími kritickými rozměry je naklonění těžce osy válcové vazby kolem osy  $y_1$  (viz obr. 8.17) a posuv kabiny ve směru osy  $y_1$ , viz obr. 8.16.

## 8.9 Alternativní provedení pravostranného mechanismu řazení

Z důvodů uvedených v závěru kap. 8.7 bylo přistoupeno k návrhu alternativní varianty pravostranného mechanismu řazení. V této kapitole budou stručně popsány dva variantní návrhy A a B, které se však zdály být svými charakteristikami méně vhodné než třetí variantní řešení C, které je podobněji rozebráno. Všechna alternativní řešení jsou založena na strukturální změně v oblasti trámce 10, který se největší měrou podílí na nestabilitě mechanismu, viz kap. 8.8.2.

### 8.9.1 Variantní návrh A pravostranného mechanismu řazení



Obrázek 8.19: Kinematické schéma variantního návrhu A pravostranného mechanismu řazení

Strukturální změna původního mechanismu, která vede k alternativnímu návrhu A,

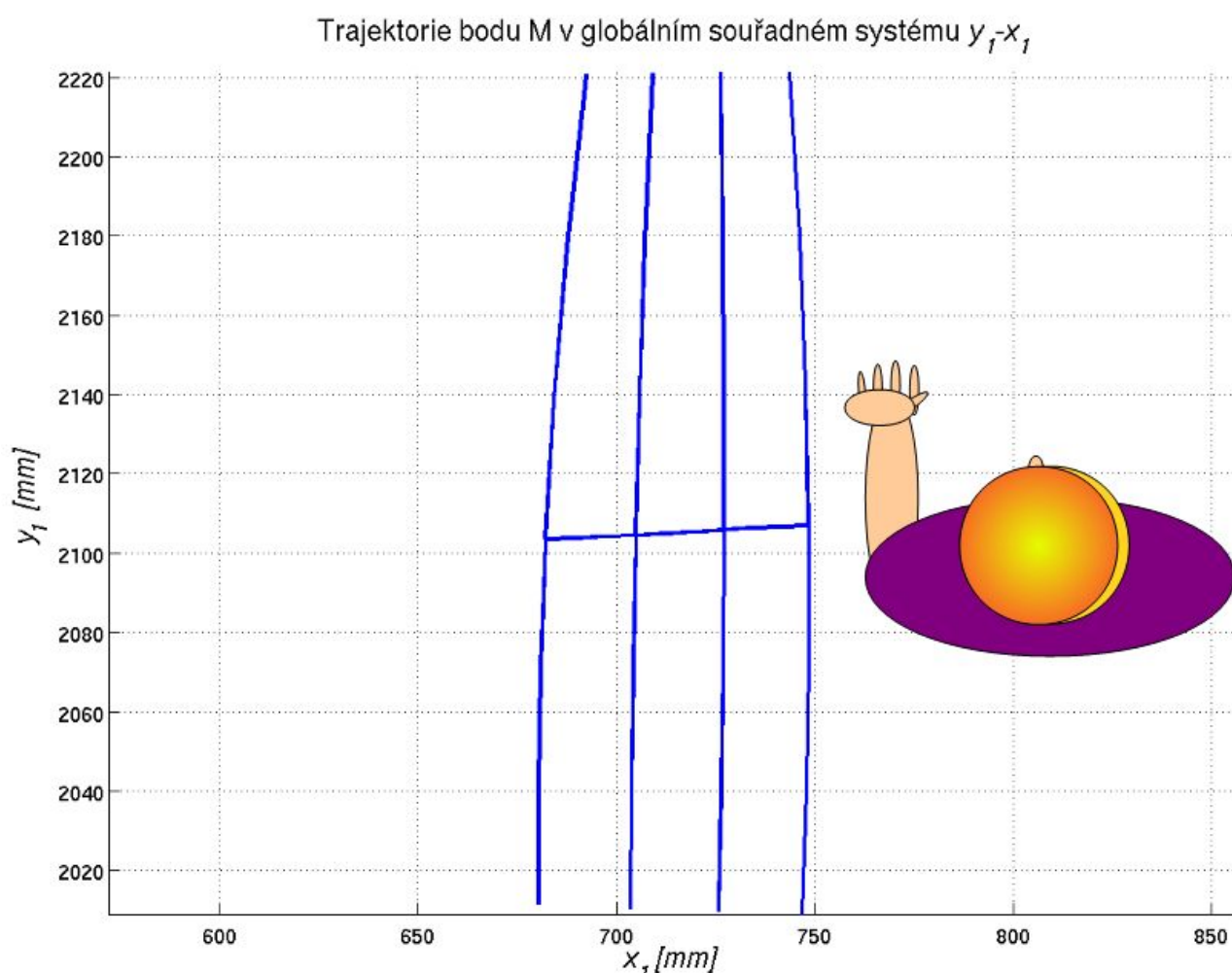
spočívá pouze ve změně polohy osy válcové vazby trámce 10 tak, aby souhlasila se směrem jízdy automobilu. Rozměry mechanismu zůstaly zachovány až na polohu osy válcové vazby trámce 10, která byla otočena o  $90^\circ$  a posunuta ve směru osy  $z_1$  o  $200\text{ mm}$  směrem nahoru. Kinematické schéma této varianty je uvedeno na obr. 8.19.

Průběh drah řazení je uveden v obr. 8.20, pokud jej srovnáme s tvarem drah pro původní koncept, není vidět změna. Bohužel i citlivost návrhu vůči změně rozměrů je vysoká. Cílová funkce nabyla pro toto strukturální řešení hodnoty

$$F = 12 \cdot 10^{-4}. \quad (8.25)$$

Také z důvodů silových by asi tato varianta nebyla vhodným kandidátem, protože při pohybu zvaném řazení by nejspíš docházelo ke vzpříčení trámce 10, proto bylo od varianty upuštěno.

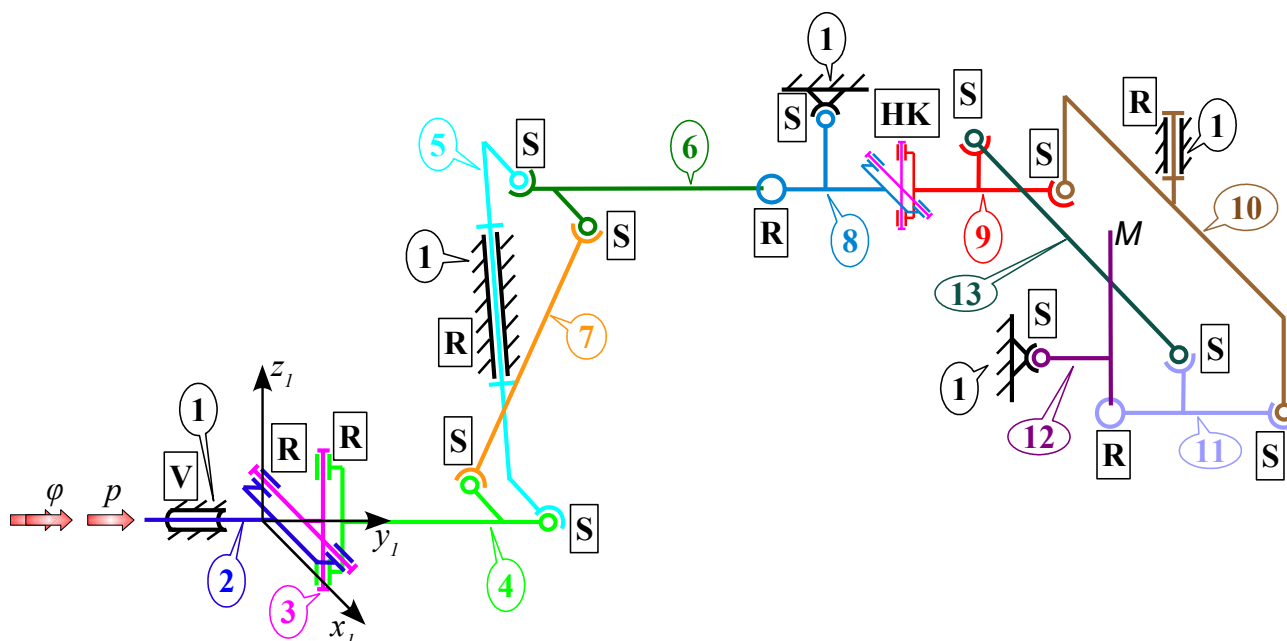
Algoritmus výpočtu úlohy polohy pravostranného mechanismu řazení varianty A a hodnoty cílové funkce napsaný v programovacím jazyku používaném v softwaru MATLAB je přiložen v 11.12.



Obrázek 8.20: Dráhy řazení alternativního návrhu A pravostranného mechanismu řazení



## 8.9.2 Variantní návrh B pravostranného mechanismu řazení



Obrázek 8.21: Kinematické schéma variantního návrhu B pravostranného mechanismu řazení

Alternativní varianta B, viz kinematické schéma z obr. 8.21, je také založena zejména na změně trámce 10, jehož válcová vazba je nahrazena vazbou rotační s vertikální osou. Aby byl přenášen pohyb volby ze členu 8 na 9 je třeba spojit tato tělesa Hookovým kloubem, který je ve schématu značen zkratkou HK; těleso 9 je dále vázáno se členem 10 pomocí sférické vazby. Abychom zajistili přenos pohybu volby na řadicí páku 12, musí být tyč 13 vázána k tělesu 9, a to pomocí sférického kloubu.

Rozměry variantního návrhu B byly zvoleny stejné s původním konceptem mechanismu.

Na následujícím obr. 8.22 je zobrazena trajektorie bodu M rukojeti řadicí páky 12. Hned na první pohled lze pozorovat značnou změnu oproti oběma předchozím konstrukčním konceptům, trajektorie je pro tento alternativní návrh B přirozenější, dráhy řazení nejsou již tak nepřírodně blízko sebe jako v případech předchozích. Tvar trajektorie by samozřejmě mohl být upraven optimalizací vhodných ladicích parametrů. Nejpodstatnější však je skutečnost, že citlivost tohoto alternativního řešení vzhledem k výrobním a montážním imperfekcím je o mnoho nižší než u původního konstrukčního návrhu. Byly zkoumány například citlivosti na změnu polohy kabiny nebo změnu polohy i natočení osy rotační vazby, která váže k rámu 1 člen 10, atd.

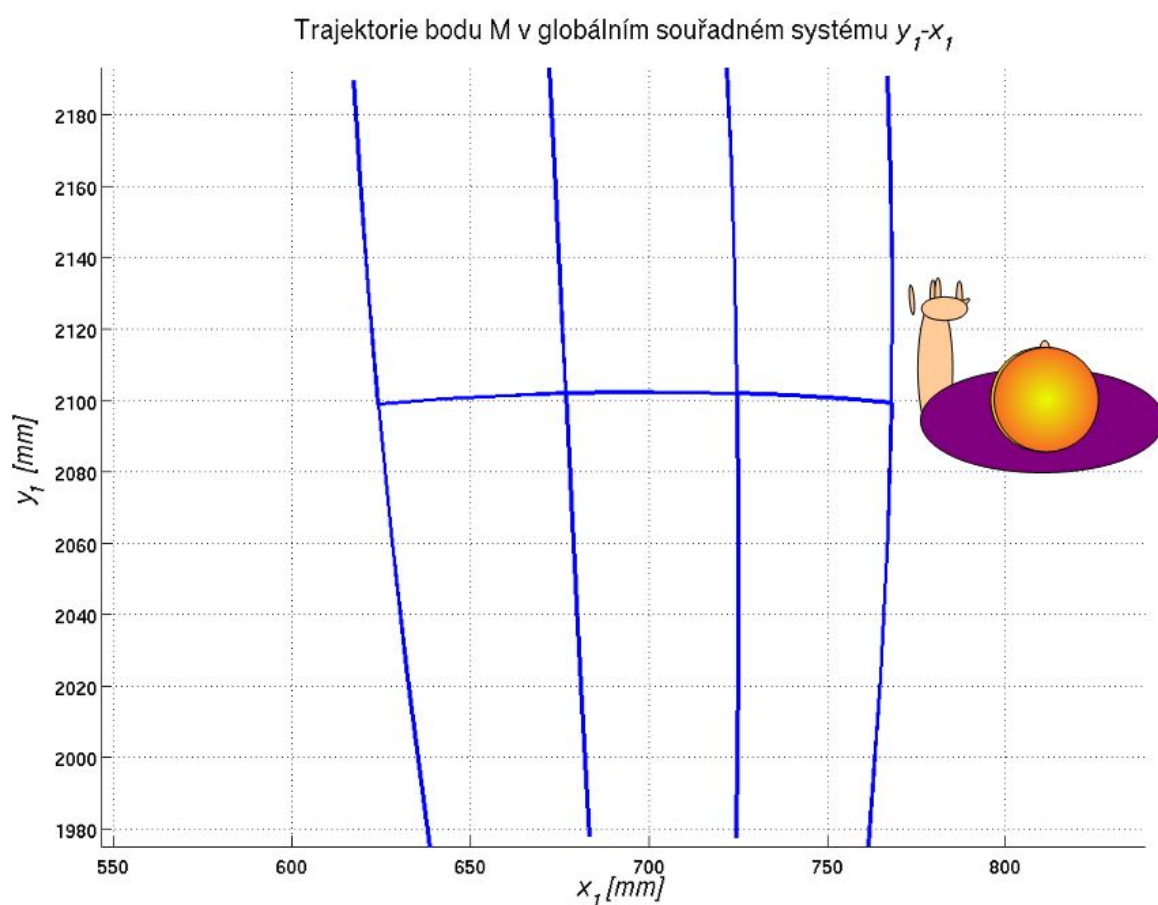
Vyskytla se však jedna komplikace, a to, že polohy rychlostních stupňů jsou obráceny okolo osy  $x_1$  (kupříkladu tam, kde má být první rychlostní stupeň, je druhý atd.); tento problém lze však jednoduše řešit hned několika způsoby (viz následující kap. 8.9.3). Tato varianta B by již

mohla sloužit jako podklad pro praktické řešení, jejíž velkou výhodou je, že změny ve struktuře nejsou tak markantní jako v variantním provedení C, takže by výrobce mohl použít více stávajících dílů.

Hodnota cílové funkce tohoto alternativního řešení je

$$F = 34 \cdot 10^{-4}. \quad (8.26)$$

Algoritmus výpočtu úlohy polohy pravostranného mechanismu řazení varianty B a hodnoty cílové funkce napsaný v programovacím jazyku používaném v softwaru MATLAB je přiložen v 11.13.

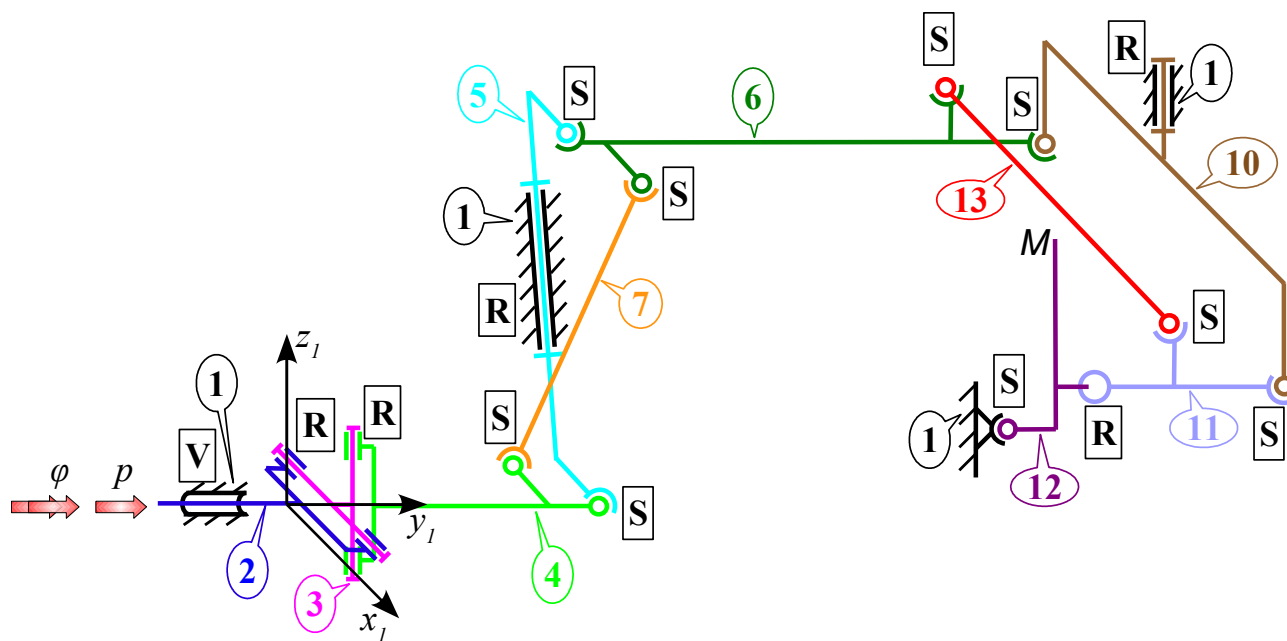


Obrázek 8.22: Dráhy řazení alternativního návrhu B pravostranného mechanismu řazení

### 8.9.3 Alternativní provedení C pravostranného mechanismu řazení

Struktura alternativního provedení C je založena na variantě B z minulé kap. 8.9.2. Po hlubším prozkoumání funkce mechanismu byl z kinematického hlediska shledán člen 8 (v případě použití rotační vazby mezi tělesem 10 a rámem 1) zbytečným a neplnícím žádnou kinematickou funkci. Proto byl z mechanismu odstraněn a tudíž členy 6 a 9 mohou být pevně spojeny v jedno těleso, které bude dále označováno pořadovým číslem 6. Z důvodu přehlednosti jsou pořadová čísla

členů zachována z původní varianty, kinematické schéma alternativního provedení je uvedeno v obr. 8.23. Komplikace zmíněná již výše s převrácením polohy rychlostních stupňů byla vyřešena posunutím rotační vazby spojující člen 11 s řadicí pákou 12 nad sférickou vazbu, která váže řadicí páku 12 k rámu 1. Dalším možným řešením je umístění jednoho ze sférických kloubů tělesa 5 na opačnou stranu než je uvedeno ve schématu.



Obrázek 8.23: Kinematické schéma alternativního provedení C pravostranného mechanismu řazení

Alternativní provedení C bylo též vymodelováno v softwarovém prostředí Pro\ENGINEER. V modulu „motion“ tohoto softwaru byla vytvořena videosekvence pohybujícího se návrhu C, která je přiložena na CD.

### Počet stupňů volnosti a základních smyček alternativního provedení C pravostranného mechanismu řazení

Počet stupňů volnosti je roven čtyřem

$$i=6 \cdot (n-l)-3 \cdot S-4 \cdot V-5 \cdot R=6 \cdot (11-l)-3 \cdot 9-4 \cdot 1-5 \cdot 5=4, \quad (8.27)$$

z nichž dva přísluší nedefinovanému otáčení tyčí 7 a 13 kolem své podélné osy a dva souřadnicím mechanismu  $p$  a  $\varphi$ .

Vypuštěním členu 8, který byl vázán k rámu 1, se snížil počet základních smyček na pět. Počet neznámých souřadnic vazeb, které budou určovány úlohou polohy, je

$$r \cdot 6 = 30, \quad (8.28)$$

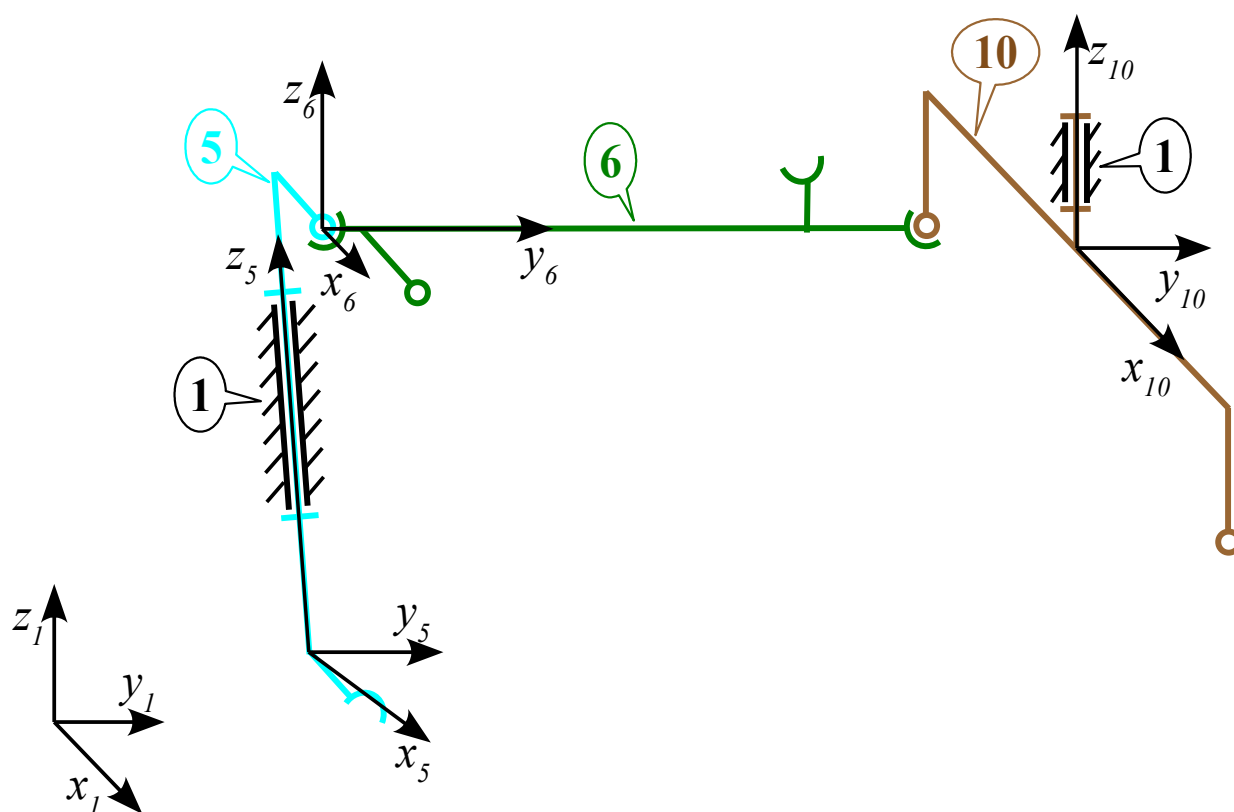
kde  $r = 5$  je počet základních smyček.

### Maticová rovnice základní smyčky I a II

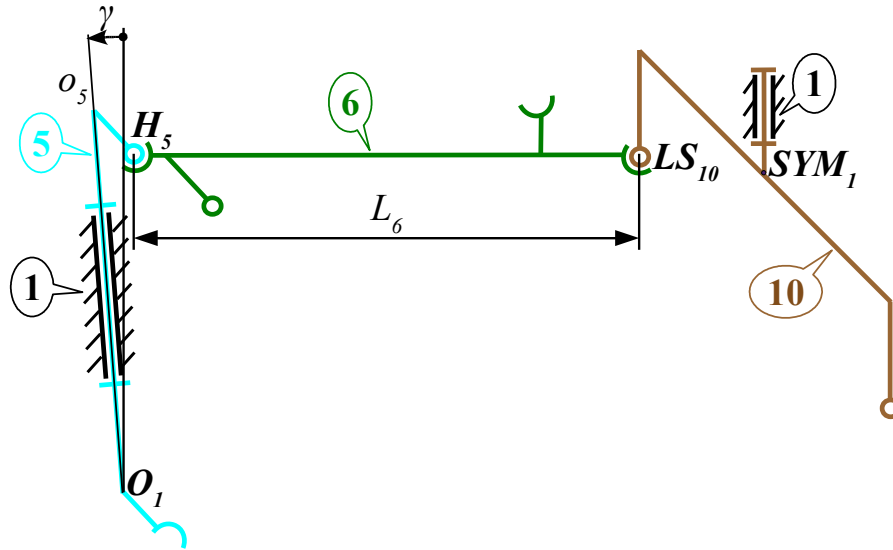
Základní smyčky I a II zůstaly oproti původnímu konstrukčnímu návrhu nezměněny včetně jejich rozměrů a maticových rovnic, viz kap. 8.4.1 a 8.4.2.

### Maticová rovnice základní smyčky III

Na obr. 8.24 a 8.25 jsou vyznačeny polohy tělesových souřadných systémů a rozměry nezbytné k popisu základní smyčky III, bod  $O_I$  je umístěn ve středu souřadného systému 5 tělesa 5, bod  $H_5$  a  $LS_{10}$  leží ve středu sférické vazby a bod  $SYM_I$  v počátku tělesového souřadného systému 10 a současně na ose rotační vazby. Úhel  $\gamma$  je vynesena za náčrtu; podrobný popis polohy tohoto úhlu lze nalézt v kap. 8.4.2.



Obrázek 8.24: Zavedení tělesových souřadných systémů potřebných při sestavování maticových rovnic základní smyčky III



Obrázek 8.25: Rozměry potřebné při sestavování maticových rovnic základní smyčky III

Maticová rovnice pro základní smyčku III bude

$$T_{15} \cdot T_{56} \cdot T_{6,10} \cdot T_{10,1} = E, \quad (8.29)$$

transformační matice je třeba vyjádřit pomocí součinu základních pohybů

$$\begin{aligned} T_{15} &= T(O_1) \cdot R_y(-\gamma) \cdot R_z(-q_6), \\ T_{56} &= T(H_5) \cdot R_x(q_7) \cdot R_y(q_8) \cdot R_z(q_9), \\ T_{6,10} &= T_y(L_6) \cdot R_x(q_{15}) \cdot R_y(q_{16}) \cdot R_z(q_{17}) \cdot T(-LS_{10}), \\ T_{10,1} &= R_z(q_{18}) \cdot T(-SYM_1). \end{aligned} \quad (8.30)$$

### Maticová rovnice základní smyčky IV

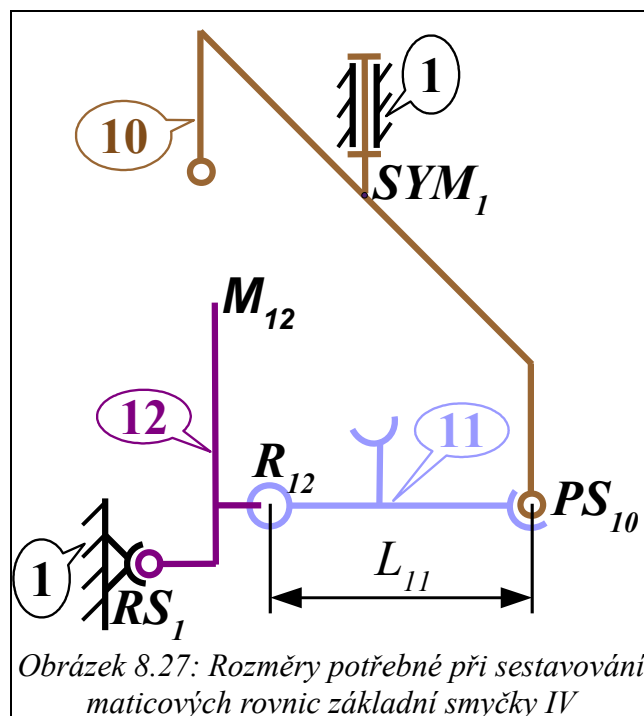
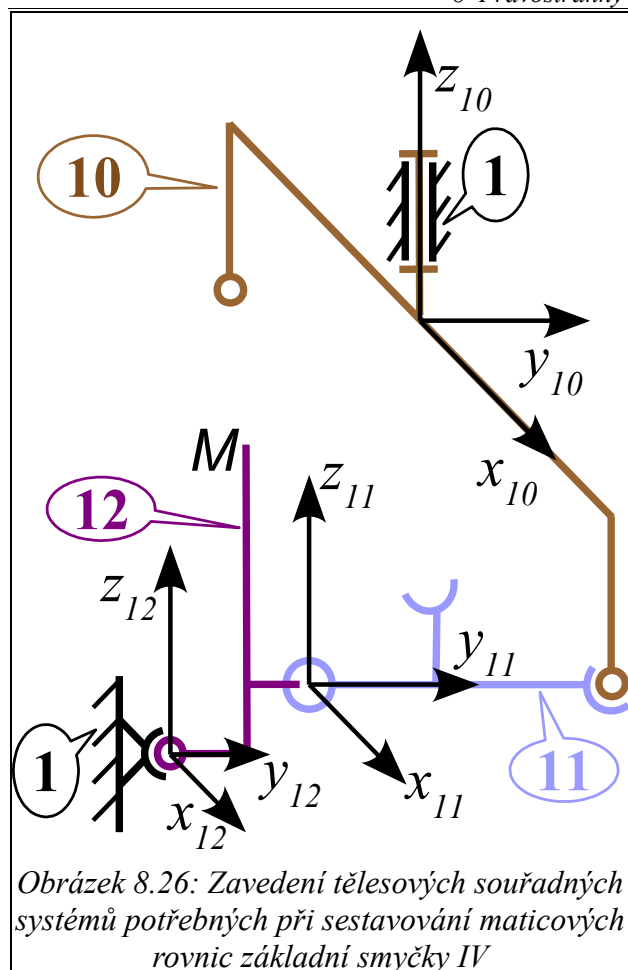
Na následujících dvou obr. 8.26 a 8.27 jsou opět vyznačeny polohy tělesových souřadných systémů a rozměry použité při popisu základní smyčky IV. Bod  $SYM_1$  byl již popsán v minulé kapitole, body  $PS_{10}$  a  $RS_1$  přísluší středům sférických kloubů, bod  $R_{12}$  leží v počátku souřadného systému 11 a současně na ose rotační vazby. Bod  $M_{12}$  je umístěn na rukojeti řadicí páky 12.

Maticová rovnice základní smyčky IV je

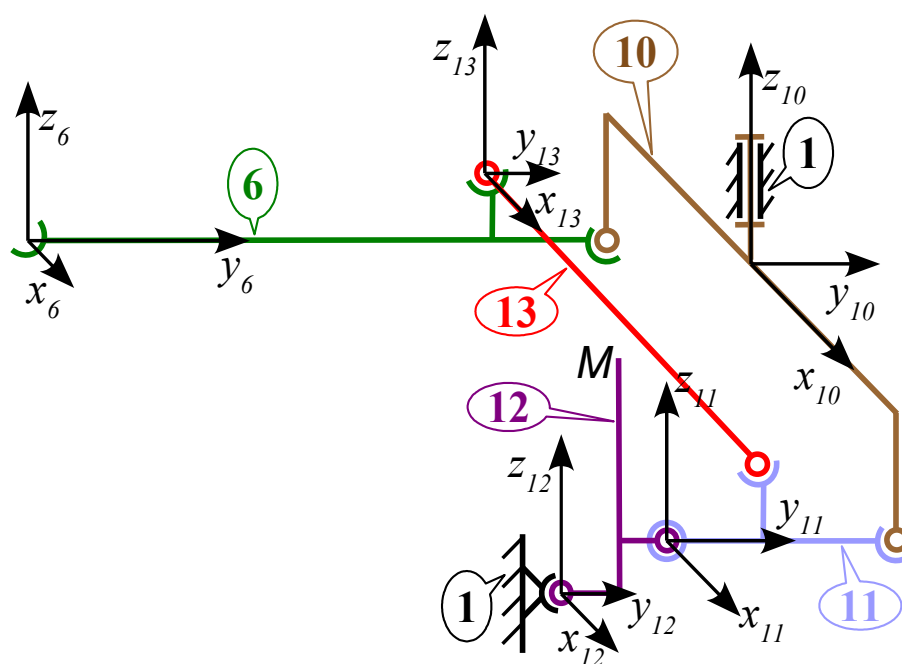
$$T_{1,10} \cdot T_{10,11} \cdot T_{11,12} \cdot T_{12,1} = E, \quad (8.31)$$

kde jednotlivé transformační matice mají tvar

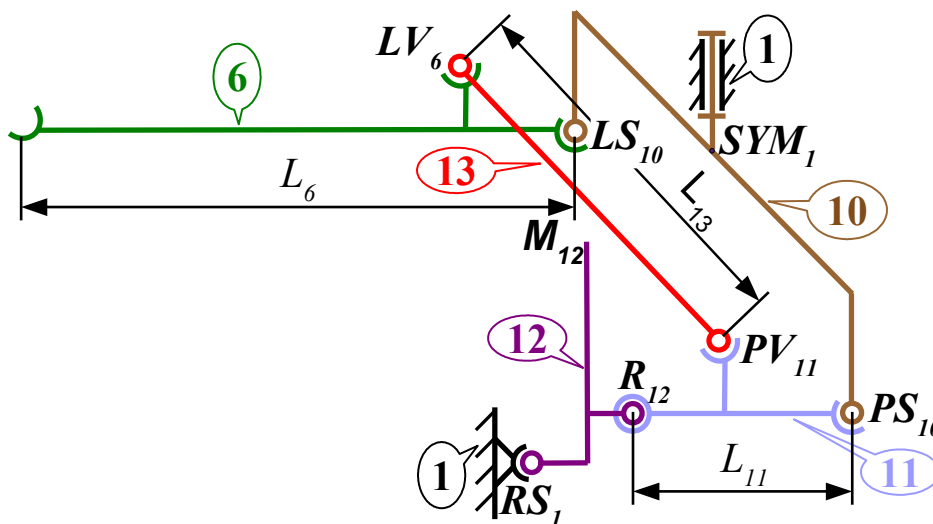
$$\begin{aligned} T_{1,10} &= T(SYM_1) \cdot R_z(-\alpha_{18}), \\ T_{10,11} &= T(PS_{10}) \cdot R_x(\alpha_{19}) \cdot R_y(\alpha_{20}) \cdot R_z(\alpha_{21}) \cdot T_y(-L_{11}), \\ T_{11,12} &= R_x(\alpha_{22}) \cdot T(-R_{12}), \\ T_{12,1} &= R_x(\alpha_{23}) \cdot R_y(\alpha_{24}) \cdot R_z(\alpha_{25}) \cdot T(-RS_1). \end{aligned} \quad (8.32)$$



### Maticová rovnice základní smyčky V



Zavedené tělesové souřadné systémy a rozměry sloužící při sestavování maticových rovnic základní smyčky V jsou vyobrazeny na obr. 8.28 a 8.29. Body  $LV_6$  a  $PV_{11}$  přísluší středům sférických kloubů, bod  $LV_6$  navíc leží v počátku tělesového souřadného systému 13.



Obrázek 8.29: Rozměry potřebné při sestavování maticových rovnic základní smyčky V

Maticová rovnice základní smyčky V bude při výše zavedených tělesových systémech

$$T_{6,10} \cdot T_{10,11} \cdot T_{11,13} \cdot T_{13,6} = E, \quad (8.33)$$

jednotlivé transformační matice je nutno rozepsat

$$\begin{aligned} T_{11,13} &= T(PV_{11}) \cdot R_x(q_{26}) \cdot R_y(q_{27}) \cdot R_z(q_{28}) \cdot T_x(-L_{13}), \\ T_{13,6} &= R_y(q_{29}) \cdot R_z(q_{30}) \cdot T(-LV_6), \end{aligned} \quad (8.34)$$

transformační matice  $T_{6,10}$  a  $T_{10,11}$  již byly uvedeny výše v 8.30 a 8.32.

### Shrnutí úlohy polohy alternativního provedení C pravostranného mechanismu řazení

Získali jsme pro pět základních smyček pět maticových rovnic 8.3, 8.5, 8.29, 8.31 a 8.33. Z každé maticové rovnice lze použít šest skalárních transcendentních rovnic, takže jsme obdrželi třicet rovnic, v nichž neznámými jsou prvky vektoru  $\mathbf{q} \in \mathbb{R}^{30}$ , které představují hodnoty úhlů natočení jednotlivých vazeb.

### Postup výpočtu trajektorie bodu M

Trajektorie bodu  $M$  rukojeti řadicí páky 12 v pevném souřadném systému 1 se vypočte stejně jako při původním konstrukčním návrhu pravostranného mechanismu řazení, viz kap. 8.4.8.

Algoritmus výpočtu úlohy polohy pravostranného mechanismu řazení varianty C a

hodnoty cílové funkce napsaný v programovacím jazyku používaném v softwaru MATLAB je přiložen v 11.14.

### Rozměry, dráhy řazení a citlivostní analýza alternativního provedení C pravostranného mechanismu řazení

Tabulka 8.5: Rozměry alternativního provedení C pravostranného mechanismu řazení

Úhlové rozměry		Souřadnice bodů	
Označení úhlu	Hodnota úhlu [°]	Označení bodu	Souřadnice bodu [mm]
$\gamma$	7	$O_1$	[-34,0; 936,0; 41,5]
Délkové rozměry		$H_5$	[61,0; 0,0; 711,5];
Označení délky	Hodnota [mm]	$LV_6$	[17,0; 1312,5; 97,5]
$L_6$	1419,0	$LS_{10}$	[-394,5; 0,0; -100,0]
$L_{11}$	265,0	$SYM_1$	[313,5; 2339,5; 809,0]
$L_{13}$	754,0	$PS_{10}$	[394,5; 0,0; -99,0]
		$PV_{11}$	[-17,0; 173,5; 97,5]
		$R_{12}$	[0,0; 5,5; 59,0]
		$RS_1$	[708,0; 2069,0; 650,5]
		$M_{12}$	[17,0; 35,0; 348,0]

Rozměry mechanismu pro základní smyčky I a II jsou uvedeny v kap. 8.6, většina rozměrů týkajících se základních smyček III, IV a V byla převzata z původního konceptu a nové rozměry byly voleny tak, aby se mechanismus pohyboval v podobném prostoru, což je důležité z hlediska zástavby nákladního automobilu. Dále došlo k naladění délky členu 6  $L_6$  tak, aby hodnota cílové funkce byla co nejmenší. Všechny rozměry použité při sestavování maticových rovnic základních smyček III, IV a V jsou shrnuty v následující tabulce.

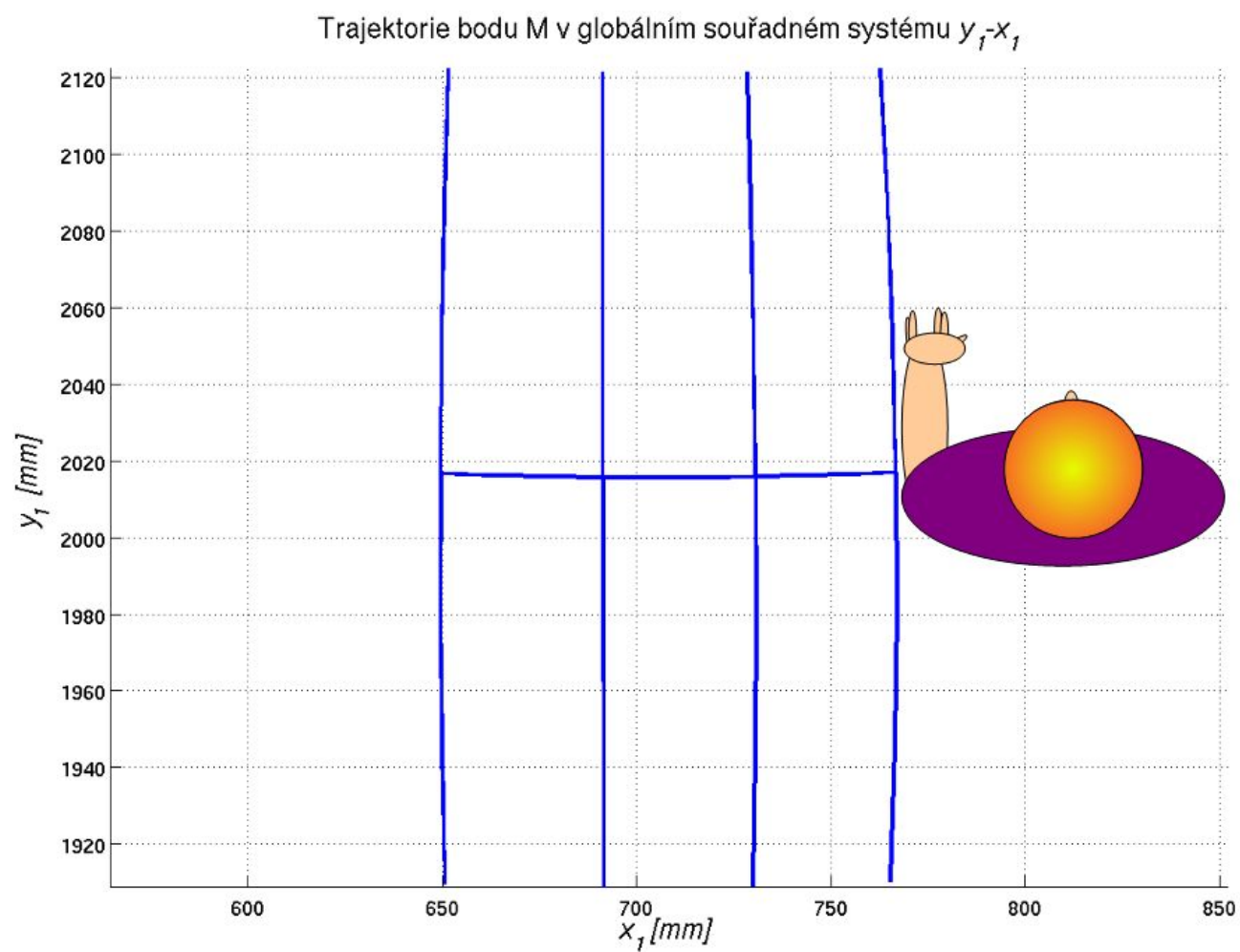
Trajektorie bodu M řadicí páky 12 v pevném souřadném systému 1 je uvedena v obr. 8.30 níže a lze prohlásit, že dráhy řazení se v tomto případě již blíží optimu, o čemž vypovídá i hodnota cílové funkce

$$F = 1,2 \cdot 10^{-4}. \quad (8.35)$$

Při srovnání s grafem původní varianty (viz obr. 8.15) je vidět markantní zlepšení tvaru drah řazení, také citlivostní analýza (vzhledem k omezenému rozsahu práce nebyla publikována) ukázala, že stabilita alternativního provedení C mechanismu vzhledem k výrobním a montážním nepřesnostem bude větší než v případě původního konceptu. Z těchto důvodů je alternativní provedení pravostranného mechanismu řazení C z hlediska kinematických vlastností shledáno jako



vhodnější než původní koncept.



Obrázek 8.30: Dráhy řazení alternativního provedení C pravostranného mechanismu řazení

## 9 Shrnutí dosažených výsledků a přínosů disertační práce

Shrnutí dosažených výsledků bude provedeno z pohledu splnění cílů disertační práce vytyčených v kap. 2.2. Cíle disertační práce vyplynuly ze zkušeností školitele a doktoranda a současně z provedené rešerše stavu poznání v oblasti analýzy a syntézy mechanismů, viz kap. 2.1.

### 9.1 Metodika analýzy složitých prostorových mechanismů

*Metodika analýzy mechanismů metodou transformačních matic je zpracována v kap. 3.* Autor se snažil podat problematiku srozumitelně a současně popsal všechna úskalí, se kterými se dosud setkal při zpracování problémů technické praxe.

Přínos a srozumitelnost metodiky nemůže autor hodnotit. Vzhledem k tomu, že požadavek na vytvoření této metodiky částečně vyplynul z impulzů od studentů předmětu Mechanika soustav těles, bude jim tato metodika dána k dispozici a od posluchačů budou získávány zpětné vazby. Podle tohoto autor doplní nebo přepracuje nesrozumitelné nebo nedostatečně rozepsané pasáže.

*Přínosem kap. 3 je navržení výpočtu matic rychlostí a zrychlení využitím numerických derivací místo použití diferenciálních operátorů, viz kap. 3.3.10.* Navržený postup výpočtu rychlostí a zrychlení řešiteli značně zjednodušuje práci a současně také snižuje pravděpodobnost toho, že se řešitel dopustí chyby. Navrhovaný postup je úspěšně ověřen v kap. 6 na prohozním mechanismu a jeho výstupy jsou srovnány s výsledky získanými z multibody programového prostředí je uvedeno v kap. 6.7.

Poznamenejme, že i syntéza mechanismů rozpracovaná v kap. 4 je publikována ve formě metodiky a bude taktéž dána k dispozici studentům. Jako součást metodiky slouží i zpracované problémy technické praxe uvedené v kap. 6, 7 a 8.

### 9.2 Návrh matematické metody pro řešení úlohy polohy mechanismů

Poloha mechanismu je dle kap. 3.2.5 popsána soustavou nelineárních, transcendentních a skalárních rovnic (viz vztah 3.4). Z tohoto důvodu a proto se kap. 5 zabývá hledáním vhodné metody pro řešení soustav nelineárních rovnic. V disertační práci bylo zkoumáno šest metod. Navržené metody byly podrobeny numerickým testům na čtyřech testovacích soustavách rovnic. *Jako nejvhodnější metoda byla vyhodnocena Martinezova hybridní metoda.* V tab. 5.6 jsou shrnuty výsledky těchto testů. Pro vyhodnocení použijeme dvě kritéria, a to spolehlivost a efektivitu. Z

tabulky je zřejmé, že nejspolehlivější metodou je Martinezova hybridní metoda, která konvergovala k řešení ve všech případech. Z pohledu efektivity jsou jasně nejlepší dvě metody, a to Broydenova dobrá metoda a Martinezova hybridní metoda. Z tohoto vyplývá, že vzhledem ke spolehlivosti i efektivitě je nejvhodnější metodou pro řešení soustav nelineárních rovnic Martinezova hybridní metoda. Je velice důležité připomenout, že na základě provedených testů (viz obr. 5.1) je zapotřebí u této metody sestavit počáteční odhad Jacobiho matice pomocí vztahů pro numerický výpočet derivace dle 5.13. Použít jako počáteční odhad jednotkovou matici se ukázalo jako kriticky nevhodné – docházelo k divergenci algoritmu.

Při porovnání efektivity námi zvolené Martinezovy hybridní metody s přednastavenou metodou ve světově uznávaném a užívaném matematickém softwaru MATLAB 7.9.0 - konkrétně je to metoda Trust-Region Dogleg - zjistíme, pokud bychom uvažovali pouze jeden výpočet úlohy polohy, tak dle 5.58 je Martinezova metoda 5.8-krát efektivnější. Vyšší efektivita zvláště nabývá na významu u optimalizačních úloh, kde je úloha polohy řešena běžně v počtech deseti tisíců.

Dalším podstatným výstupem této kapitoly jsou poznatky o chování soustav rovnic, které popisují nesestavitelný mechanismus, viz kap. 5.8.4. Bylo zjištěno, že u nesestavitelných mechanismů délka kroku  $p$  značně osciluje, a to většinou nad délkou prvního kroku, z čehož plyne důležitý závěr, že do algoritmu pro výpočet soustav nelineárních rovnic, je vhodné implementovat další ukončovací kritérium 5.55. To v praxi znamená, že algoritmus je ukončen daleko dříve než by došlo k vyčerpání maximálního počtu iterací, a proto je ušetřeno mnoho zbytečných vyčíslení funkce  $f$ . Toto kritérium opět zvláště významné u optimalizace, kdy optimalizační algoritmus běžně navrhuje rozměry mechanismu tak, že soustavu nelze sestavit a pokud se tak stane, je potřeba co nejdříve ukončit algoritmus pro výpočet soustavy rovnic a přejít na speciální postup popsany v 4.6.

### 9.3 Implementace optimalizačních metod do úloh syntézy mechanismů

Užití metod pro syntézu mechanismů je popsáno v kap. 4. V úvodu této části textu je definována formulace optimalizačního problému při syntéze mechanismů na jednoduchém příkladu a následovně je tato formulace zobecněna tak, jak je uváděno v odborné matematické literatuře, aby byl čtenář seznámen s terminologií a mohl případně využít i jiné metody než ty, které jsou zde prezentovány. *Velmi významný je závěr kap. 4.1, kde je uvedena charakteristika cílových funkcí a omezovacích podmínek, protože dle těchto charakteristik je nutno vybírat vhodné metody optimalizace pro syntézu mechanismů.*

V kap. 4.2 je popsána citlivostí analýza, která je nezbytná při přípravě optimalizace mechanismu a lze ji také využít v technické praxi pro stanovení výrobních a montážních tolerancí mechanismu.

*Pro optimalizační úlohy bez omezení jsou navrženy dvě metody, a to metoda simplexů a metoda BFGS, viz kap. 4.3. Metody byly vybrány na základě rozboru charakteristiky cílové funkce. Metoda simplexů patří k těm, které nevyužívají derivace cílové funkce, což je u syntézy mechanismů zvláště důležitá vlastnost optimalizační metody, protože nejsme schopni analyticky vyjádřit derivaci a její numerické vyjádření je výpočetně drahé. Nevýhodou metody je, že nijak nevyužívá historii, tzn. nijak nevyužívá hodnot cílové funkce, které byly dříve vypočteny. Druhá navržená metoda optimalizace bez omezení je metoda BFGS, jejíž přednosti a nedostatky jsou přesně opačné než u předchozí metody. Výhodou metody je, že využívá historii, tzn. že pomocí vypočtených hodnot cílových funkcí staví kvadratický model cílové funkce. Metoda BFGS se v testech ukázala efektivnější než metoda simplexů.*

*Navrženou metodou pro optimalizaci s omezovacími podmínkami je metoda penalizace, která je použitelná pro všechny druhy omezovacích a jednoduchých. Svojí jednoduchostí umožňuje uživateli její plné pochopení a tedy i správné ovládání. Více k metodě penalizace je uvedeno v kap. 4.4.*

Efektivita a testy výše uvedených algoritmů jsou diskutovány v následující kapitole.

## 9.4 Problémy technické praxe

### Prohozní mechanismus

Jak již bylo uvedeno výše, v kap. 6 popisující *dynamickou úlohu prohozního mechanismu tkacího stroje* byl úspěšně ověřen navržený postup výpočtu rychlostí a zrychlení mechanismu pomocí matice převodu transformační matice a derivace matice převodu, viz kap. 3.3.10.

### Levostranný mechanismus řazení

Kinematické schéma levostranného mechanismu řazení včetně vyznačených rozměrů je uvedeno na obr. 7.10. U tohoto mechanismu se výrobce potýkal se silně zakřivenými drahami řazení dle 7.11 což by pro řidiče vozu znamenalo značné potíže při řízení. Vzhledem ke značné strukturální složitosti mechanismu nebyl výrobce schopen navrhnout mechanismus, který by dosáhl ideální trajektorie madla řadicí páky. *Mechanismus byl optimalizován tak, aby došlo k dosažení*

ideální trajektorie řazení dle obr. 7.11. Provedená citlivostní analýza ukázala, které rozměry mechanismu vhodné pro optimalizaci, viz obr. 7.13.

Dle přání zadavatele byly provedena dvourozměrná optimalizace bez omezení ladicích parametrů  $L_{ds}$ ,  $L_{os}$ , a to metodou simplexů a BFGS, viz kap. 7.9.1 a 7.9.2.

V kap. 7.9.7 jsou srovnány výsledky provedené třírozměrné optimalizační úlohy bez omezení metodou simplexů, metodou BFGS, mutibody prostředím ANSYS a ADAMS. Výsledná data optimalizačních procesů jsou porovnána v tab. 7.7. Všimněme si, že každá metoda našla odlišný výsledný vektor ladicích parametrů, z čehož plyne, že existuje několik lokálních minim cílové funkce a tudíž terén této funkce bude dosti komplikovaný.

*Z pohledu efektivity i časové náročnosti je na tom nejlépe výpočet provedený autorem metodou BFGS. Optimalizace provedená v prostředí ADAMS má druhý nejlepší výpočtový čas i výsledek. Připomeňme však, že příprava optimalizace v tomto softwaru byla značně náročná na uživatele. Z hlediska výpočtového času zcela propadá software ANSYS, který však je potřeba ocenit za velice příjemné a intuitivní ovládání. Metodu simplexů lze pro její robustnost doporučit pro případ, že by metoda BFGS havarovala. Vzhledem k její definici nutně musí být méně efektivní než metoda BFGS.*

Je nezbytné připomenout, že takto krátké optimalizační časy autorových výpočtů jsou důsledkem efektivního výpočtu úlohy polohy a tudíž výsledkem výzkumu prezentovaného v kap. 5. Pokud bychom použili přednastavené řešiče softwaru MATLAB, byly by výpočty dle 7.39 27-krát pomalejší.

Vzhledem k tomu, že optimalizace bez omezení navrhovala ladicí parametry tak, že zasahovaly do zástavby vozu, byla provedena pro tento mechanismus také optimalizace s omezovacími podmínkami, která zohledňovala zástavbu nákladního vozidla. V tab. 7.10 jsou srovnány výsledky optimalizace s omezujícími podmínkami provedené metodou penalizace ve spojení s metodou BFGS, mutibody prostředím ANSYS a ADAMS. Výsledky získané autorem metodou penalizace a ze softwaru ADAMS byly srovnatelné. Mírně slabší výsledky byly získány z prostředí ANSYS. Výpočtový čas je opět zdaleka nejhorší u prostředí ANSYS a nejrychlejší se ukázala penalizační metoda.

Všechny navržené varianty byly předány zadavateli, který vybral jednu z nich a která je úspěšně používána na nákladním voze v praxi.

## Pravostranný mechanismus řazení

Kinematické schéma pravostranného mechanismu řazení je uvedeno a obr. 8.4. Po konzultaci s výrobcem byl shledán vypočtený tvar drah řazení uspokojivým, ale bohužel tvar drah na vyrobených nákladních automobilech neodpovídal tvaru vypočtenému a navíc se lišil pro každý vyrobený automobil. Tento jev lze vysvětlit pouze přítomností výrobních a montážních imperfekcí, které způsobí, že pro každý vyrobený kus jsou dráhy řazení jiné. Cesty, jak vyřešit vzniklý problém, jsou dvě. První cestou bylo identifikovat rozměry, na jejichž změnu je mechanismus citlivý. Druhou možností je změna struktury mechanismu tak, aby stabilita variantního návrhu vzhledem k vzniklým výrobním a montážním odchylkám byla větší.

*Z důvodů identifikace problematických rozměrů mechanismu a následného předepsání výrobních a montážních tolerancí byla provedena v kap. 8.8 citlivostní analýza, která potvrdila předchozí domněnky o nestabilním chování mechanismu. Byly nalezeny a vyhodnoceny výrobní a montážní tolerance kritických rozměrů.*

*V práci byly také navrženy úpravy struktury pravostranného mechanismu tak, aby bylo chování mechanismu odolnější vůči nepřesnostem výroby a montáže. Tři vypracované variantní návrhy jsou prezentovány v kap. 8.9. Kinematické schéma nejvhodnějšího alternativního provedení C je vyobrazeno na obr. 8.23. Citlivostní analýza ukázala, že stabilita alternativního provedení mechanismu vzhledem k výrobním a montážním nepřesnostem je značně větší než v případě původního konceptu.*

## 9.5 Algoritmy

*Dalším neopomenutelným přínosem práce jsou výpočetní algoritmy popsané v práci. Všechny tyto algoritmy jsou ověřené autorem a tudíž funkční. V přílohách jsou uvedeny tyto algoritmy naprogramované v programovacím jazyku matematického prostředí MATLAB. Dále jsou přiloženy procedury, které slouží k výpočtu úlohu polohy, cílové funkce a optimalizace mechanismů. Struktura těchto procedur je při použití prezentované metodiky lehce modifikovatelná na jakýkoliv typ mechanismu.*

## 9.6 Podněty k dalšímu výzkumu a práci

V práci bylo provedeno ověření navrhovaných postupů dynamického výpočtu a hlavně syntézy mechanismů pouze na jednom mechanismu. Původním plánem autora bylo ověřit navrženou metodu syntézy také na mechanismu řízení, který slouží k ovládání natočení kol vozidla.

Konkrétní typ mechanismu sloužil pro vůz se čtyřmi nápravami. Toto již nebylo provedeno z důvodu již tak značného rozsahu práce.

Dalším budoucím úkolem autora je publikovat návod k provedení syntézy mechanismů v multibody prostředích ADAMS a ANSYS. Zvláště u prvního prostředí je absolutní nedostatek takového textu a orientace v softwaru je obtížná.

Posledním dlouhodobým výhledem autora je rozšířit provedené syntézy na dynamické úlohy s uvážením elastických členů. Zde by již bylo potřebovat implementovat do výpočtů analýzu deformace součástí pomocí metody konečných prvků.

## 10 Conclusion

Contributions and aims of the dissertation thesis can be sort into the four major issues, as follows:

- The methodology of mechanisms analysis by the method of transformation matrices
- The numerical method for solving mechanisms position
- Optimization methods used for synthesis of mechanisms
- Problems from practice

The methodology of mechanisms analysis is published in the first chapter. The focus on intelligibility. Introduced terms and methods are illustrated on simple problems. The author covered all complications occurring during analysis process. The methodology will be given to students and feedbacks will be collected by the author. New method of velocity and acceleration computing is suggested in the work. The method is established on numerical derivatives.

Mechanism position is represented by system of nonlinear equations. Six methods for solving these systems were tested. Martinez hybrid method has proved highest efficiency and stability.

Optimization methods are suggested for mechanisms synthesis. Simplex method and BFGS method is suggested for solving unconstrained optimization. BFGS method was more efficient. On the other hand, simplex method has shown as more robust. Penalty method recommended for constrained optimization problems in the thesis. Remarkable advantages of the method is universal applicability and simplicity so investigator can fully understand the optimization process.

Three problems from practice conclude the work. The problems describe pitching mechanism and lefthand and righthand shifting mechanisms.

In case of lefthanded shifting mechanism, the aim was to optimize the trajectory of a shifting lever handgrip in order to obtain an acceptable ergonomic trajectory. A mathematical surrogate of the left-hand shifting mechanism has been constructed. The use of the optimization algorithms resulted in four modifications of mechanism dimensions. One of the suggested modifications has been accepted by the truck producer and has been put into operation.

The shifting lever handgrip trajectory of the righthand mechanism has been found



acceptable. Unfortunately practical experiences showed that the each piece of the produced truck has a different trajectory. Therefore sensitivity analyses have been provided in order to identify the critical dimensions the change of which can cause the distortion of the trajectory shape. Three variants of structural changes have been designed since the producer cannot keep the critical dimensions of mechanism in a prescribed bounds. Two of these variants possess better properties than the original one.

Enclosed algorithms and procedure are also valuable contributions of the thesis.

## Seznam použitých pramenů

- [1] MCPHEE, J. *Multibody System Dynamics: Research Activities* [online]. 2011-9-12 [cit. 2011-11-09]. Dostupné z WWW: <<http://real.uwaterloo.ca/~mbody/#Software>>.
- [2] CUADRADO, J. *IFTOMM Technical Committee for Multibody Dynamics* [online]. 2010 [cit. 2011-11-09]. Dostupné z WWW: <<http://iftomm-multibody.org/>>.
- [3] DUKKIPATI, R. V. *Spatial Mechanisms : Analysis and Synthesis*. Boca Raton : CRC Press, 2001. 363 s. ISBN 0849309328.
- [4] MALLIK, A. K.; GHOSH, A.; DITTRICH, G. K. *Kinematic Analysis and Synthesis of Mechanisms*. Boca Raton : CRC Press, 1994. 688 s. ISBN 9780849391217.
- [5] JULIŠ, K., et al. *Mechanika I. díl : Statika a kinematika*. Praha : SNTL, 1986. 477 s.
- [6] BRÁT, V. *Maticové metody v analýze a syntéze prostorových vázaných systémů*. Praha : Academia, 1981. 212 s.
- [7] LEDERER, P. *Teorie a optimalizace mechanických systémů I*. Praha : České vysoké učení technické v Praze, 1988. 143 s.
- [8] LEDERER, P. *Teorie a optimalizace mechanických systémů II*. Praha : České vysoké učení technické v Praze, 1989. 70 s.
- [9] ŽMINDÁK, M., et al. *Optimalizácia mechanických sústav*. Žilina : EDIS ŽU, 200. 261 s.
- [10] SÁGA, M., et al. *Aplikácia optimalizačných algoritmov v mechanike telies*. Žilina : VTS pri ŽU, 2006. 240 s. ISBN 80-969165-9-9.
- [11] BOCKO, J.; DELYOVÁ, I. *Optimalizácia mechanických sústav*. Košice : Technická univerzita v Košiciach, Strojnícka fakulta, 2010. 105 s. ISBN 978-80-553-0346-8.
- [12] JENSEN, O. F.; HANSEN, J. M. *Dimensional Synthesis of Spatial Mechanisms and the Problem of Non-Assembly*. *Multibody System Dynamics*. 2006, 15, 2, s. 107-133. ISSN 1384-5640.
- [13] HANSEN, J. M. *Synthesis of Mechanisms*. In *Advanced Design of Mechanical Systems: From Analysis to Optimization* [online]. Vienna : Springer, 2009 [cit. 2011-01-13]. Dostupné z WWW: <<http://www.springerlink.com/content/k3uwl6q5488768lk/fulltext.pdf>>. ISBN 978-3-211-99461-0.
- [14] BRÁT, V; JÁČ, V; ROSENBERG, J. *Kinematika*. Vyd. 1. Praha : SNTL, 1987. 250 s.
- [15] NOCEDAL, J.; WRIGHT, S. J. *Numerical Optimization*. New York : Springer, 2000. 636 s. ISBN 0-387-98793-2.
- [16] CHARVÁT, J. *Syntéza mechanismů : Část I*. Liberec : Vysoká škola strojní a textilní v Liberci, 1966. 154 s.
- [17] SUN, W.; YUAN, Y. *Optimization Theory and Methods Optimization Theory and Methods : Nonlinear Programming*. New York : Springer, 2006. 687 s. ISBN 978-0-387-24975-9.
- [18] FLETCHER, R. *Practical Methods of Optimization*. Cornwall : Wiley, 2000. 450 s. ISBN 978-0-471-49463-8.
- [19] RARDIN, R. L. *Optimization in Operations Research*. New Jersey : Prentice Hall, 2000. 919 s. ISBN 0-02-398415-5.

- [20] RAO, S. S. *Engineering Optimization : Theory and Practice*. Fourth Edition. New Jersey : Wiley, 2009. 813 s. ISBN 978-0-470-18352-6.
- [21] JEFFREY, C, et al. CONVERGENCE PROPERTIES OF THE NELDER-MEAD SIMPLEX METHOD IN LOW DIMENSIONS. *Journal of Society for Industrial and Applied Mathematics* [online]. December 1998, vol. 9, no. 1 , [cit. 2011-11-16]. Dostupný z WWW: <[http://www.aoe.vt.edu/~cliff/aoe5244/nelder\\_mead\\_2.pdf](http://www.aoe.vt.edu/~cliff/aoe5244/nelder_mead_2.pdf)>.
- [22] YANG, W. Y., et al. *Applied Numerical Methods Using MATLAB*. New Jersey : Wiley, 2005. 528 s. ISBN 978-0-471-69833-3.
- [23] PHUA, P; ZENG, Y. Parallel Quasi-Newton Algorithms for Large-Scale Optimization. *In Technical Reports* [online]. Singapore : NUS, 1995 [cit. 2011-11-16]. Dostupné z WWW: <<http://hdl.handle.net/1900.100/1380>>.
- [24] GOULD, N. *An introduction to algorithms for continuous optimization* [online]. Oxford : Oxford University Press, 2006 [cit. 2011-11-16]. Dostupné z WWW: <<http://www.numerical.rl.ac.uk/nimg/oupartc/lectures/paper/paper.pdf>>.
- [25] MINNAAR, R. J.; TORTORELLI, D. A.; SNYMAN, J. A. On nonassembly in the optimal dimensional synthesis of planar mechanisms. *Structural and Multidisciplinary Optimization*. 2001, 21, 5, s. 345-354.
- [26] HANSEN, J. M. Synthesis of Mechanisms Using Time-Varying Dimensions. *Multibody System Dynamics*. 2002, 7, 1, s. 124-144.
- [27] BARTSCH, H. J. *MATEMATICKÉ VZORCE*. Praha : SNTL, 1983. 832 s.
- [28] BROYDEN, C. G. A class of method for solving nonlinear simultaneous equations. *Mathematics of computation*. 1965, 19, 92, s. 577-593.
- [29] SHERMAN, J.; MORRISON, W. J. Adjustment of an Inverse Matrix Corresponding to Changes in the Elements of a Given Column or a Given Row of the Original Matrix. *Annals of Mathematical Statistics*. 1949, 20, 4, s. 620-624.
- [30] MARTINEZ, J. M.; OCHI, L. S. Sobre dois métodos de Broyden. *Matemática Aplicada e Computacional*. 1982, 1, s. 165-141.
- [31] LEVENBERG, K. A Method for the Solution of Certain Problems in Least-Squares. *Quarterly Applied Mathematics*. 1944, 2, s. 164-168.
- [32] MARQUARDT, D. An Algorithm for Least-squares Estimation of Nonlinear Parameters. *SIAM Journal Applied Mathematics*. 1963, 11, 2, s. 431-441.
- [33] A, GROSAN; ABRAHAM, A. A New Approach for Solving Nonlinear Equations Systems. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*. 2008, 38, 3, s. 698-714.
- [34] DOSTÁL, Z. *LINEÁRNÍ ALGEBRA*. Ostrava : VŠB - Technická univerzita Ostrava, 2000. 188 s.
- [35] NICHOLAS, J. H. EFFICIENT ALGORITHMS FOR COMPUTING THE CONDITION NUMBER OF A TRIDIAGONAL MATRIX. *SIAM J. Sci. STAT. COMPUT.*. 1986, 7, 1, s. 150-165.
- [36] Tkanina#Tkac.C3.AD stroje. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, , last modified on 13. 2. 2011 [cit. 2011-11-17]. Dostupné z WWW: <[http://cs.wikipedia.org/wiki/Tkanina#Tkac.C3.AD\\_stroje](http://cs.wikipedia.org/wiki/Tkanina#Tkac.C3.AD_stroje)>.
- [37] STEVENS, B. L.; LEWIS, F. L. *Aircraft control and simulation*. New Jersey : Wiley, 2003. 645 s.

- [38] ŠEDIVÝ, P. *Modelování pohybů numerickými metodami* [online]. Hradec Králové : MAFY, 2011 [cit. 2011-11-17]. Dostupné z WWW: <<http://fyzikalniolympiada.cz/texty/modelov.pdf>>.
- [39] VITÁSEK, E. *NUMERICKÉ METODY*. Praha : SNTL, 1987. 514 s.
- [40] EDR [online]. 1999 [cit. 2011-11-18]. ANSYS Tutorial: Design Optimization in DX - EDR. Dostupné z WWW: <[http://www.edr.no/blogg/ansys\\_bloggen/ansys\\_tutorial\\_design\\_optimization\\_in\\_dx](http://www.edr.no/blogg/ansys_bloggen/ansys_tutorial_design_optimization_in_dx)>.
- [41] ANSYS, Inc. *Baidu* [online]. 2009 [cit. 2011-11-18]. ANSYS workbench 12.0\_Introduction\_DX\_DOC. Dostupné z WWW: <<http://wenku.baidu.com/view/1c789d878762caaedd33d4f4.html>>.
- [42] ANSYS, Inc. *Design Exploration* [online]. Canonsburg : SAS, 2009 [cit. 2011-11-18]. Dostupné z WWW: <[http://www1.ansys.com/customer/content/documentation/121/wb\\_dx.pdf](http://www1.ansys.com/customer/content/documentation/121/wb_dx.pdf)>.
- [43] MSC Software. *Getting Started Using Adams/View - MSC Adams 2011* [online]. Santa Ana : MSC SimCompanion, 04/27/2011 [cit. 2011-11-18]. Dostupné z WWW: <[http://www.am.chalmers.se/~anbo/mekfk/view\\_gs.pdf](http://www.am.chalmers.se/~anbo/mekfk/view_gs.pdf)>.
- [44] REKTORYS, K, et al. *PŘEHLED UŽITÉ MATEMATIKY I*. Praha : SNTL, 1988. 607 s.
- [45] ZELINKA, I. *Umělá inteligence v problémech globální optimalizace*. Praha : BEN – technická literatura, 2002. 192 s. ISBN 80-7300-069-5.
- [46] GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Massachusetts : Addison-Wesley Publishing Company, 1989. 408 s. ISBN 83-204-2272-8.
- [47] MICHALEWICZ, Z. *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin : Springer-Verlag, 1996. 430 s. ISBN 83-204-2368-6.
- [48] SLAVÍK, J.; STEJSKAL, V.; ZEMAN, V. *Základy dynamiky strojů*. Praha : Vydavatelství ČVUT, 1997. 319 s. ISBN 80-01-01622-6.
- [49] LEDERER, H.; LONIJ, G.; GORVES, B. Application of higher order derivatives in the synthesis of crank and cam mechanisms. *Computational Kinematics* [online]. 2009, , [cit. 2011-01-13]. Dostupný z WWW: <<http://www.springerlink.com/content/gqr810g4m38lv10q/fulltext.pdf>>.
- [50] MARTÍNEZ-ALFARO, H. Four-bar Mechanism Synthesis for n Desired Path Points Using Simulated Annealing. In *Advances in Metaheuristics for Hard Optimization* [online]. Berlin : Springer, 2007 [cit. 2011-01-13]. Dostupné z WWW: <<http://www.springerlink.com/content/gqr810g4m38lv10q/fulltext.pdf>>. ISBN 978-3-540-72960-0.
- [51] KIM, Y. Y., et al Configuration Design of Rigid Link Mechanisms by an Optimization Method: A First Step. In *IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials* [online]. Netherlands : Springer, 2006 [cit. 2011-01-13]. Dostupné z WWW: <<http://www.springerlink.com/content/l2l1um530888j12x/fulltext.pdf>>. ISBN 978-1-4020-4752-7.
- [52] RAI, A. K.; SAXENA, A.; MANKAME, N. D. Unified synthesis of compact planar path-generating linkages with rigid and deformable members. *Structural and Multidisciplinary Optimization* [online]. 2002, 6, [cit. 2011-01-13]. Dostupný z WWW: <<http://www.springerlink.com/content/n696j0w8752477h6/fulltext.pdf>>. ISSN 1615-1488.

- [53] STOLPE, M.; ACHTZIGER, W.; KAWAMOTO, A. A Concept for Global Optimization of Topology Design Problems. *In IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials* [online]. Netherlands : Springer, 2006 [cit. 2011-01-13]. Dostupné z WWW: <<http://www.springerlink.com/content/l855x46768527317/fulltext.pdf>>. ISBN 978-1-4020-4752-7.
- [54] EBERHARD, P.; SEDLACZEK, K. Grid-Based Topology Optimization of Rigid Body Mechanisms. *In Advanced Design of Mechanical Systems: From Analysis to Optimization* [online]. Vienna : Springer, 2009 [cit. 2011-01-13]. Dostupné z WWW: <<http://www.springerlink.com/content/t18tu017v8631214/fulltext.pdf>>. ISBN 978-3-211-99461-0.
- [55] PODEŠVA, J. *Kinematika : soubor přednášek*. Ostrava : Vysoká škola báňská-Technická univerzita, 2001. 253 s. ISBN 80-7078-880-1.

## Seznam publikační činnosti doktoranda

FRANTIŠEK, O. Influence of Mounting and Producing Imperfections on Kinematical Properties of Mechanism. *TRANSACTIONS of the VŠB – Technical University of Ostrava, Mechanical Series*. 2009, Vol. LV, No. 1652. pp. 87-92. ISBN 978-80-248-2051-4, ISSN 1210-0471.

FRANTIŠEK, O. Analysis and Optimization of the Shifting Mechanisms. *Applied Mechanics 2009*. Bratislava: 2009. pp. 33-34. ISBN: 978-80-89313-32- 7.

FRANTIŠEK, O. Optimization of Kinematic Properties of Mechanism. *TRANSACTIONS of the VŠB – Technical University of Ostrava, Mechanical Series*. 2009, Vol. LV, No. 1653. pp. 93-100. ISBN 978-80-248-2051-4, ISSN 1210-0471.

FERFECKI, P.; FRANTIŠEK, O. COMPARISON OF METHODS FOR COMPUTATION OF ELECTROMAGNETIC FORCES IN THE RADIAL ACTIVE MAGNETIC BEARING. *TRANSACTIONS of the VŠB – Technical University of Ostrava, Mechanical Series*. 2009, pp. 51-58. ISSN 1210-0471.

FRANTIŠEK, O. Analysis and Optimization of the Shifting Mechanisms, *Engineering Mechanics 2010*. Book of extended abstracts., pp.58-59, ISBN 978-80-86246-35-2.

SROVNAL, V, et al. Embedded System Design for Health Supervisory Systems. *Ubiquitous Computing and Communication Journal* [online]. 2011, přijato-bude vydáno, [cit. 2011-05-20]. Dostupný z WWW: <<http://www.scribd.com/doc/46190320/Embedded-System-Design-for-Health-Supervisory-Systems-Ubiquitous-Computing-and-Communication-Journal>>.

Užitný vzor - Mechanický polohovatelný stůl- O. František (50%) et al.

## Životopis doktoranda

- Příjmení:* František
- Jméno:* Ondřej
- Narozen:* 2. 8. 1984 v Prostějově
- Trvalé bydliště:* Gen. Sochora 1204/9, 708 00 Ostrava-Poruba
- Státní příslušnost:* Česká republika
- Rodinný stav:* svobodný
- Telefon, email:* +420 608 876 862, [ondrej.frantisek@gmail.com](mailto:ondrej.frantisek@gmail.com)
- Vzdělání:* 1999 - 2003: Střední průmyslová škola a Střední odborné učiliště Prostějov, Lidická 4, obor: Mechanik strojů a zařízení, ukončeno maturitou s vyznamenáním  
2003 - 2006: Vysoká škola báňská - Technická univerzita Ostrava, bakalářské studium, obor: Aplikovaná mechanika, ukončeno s vyznamenáním  
2006 - 2008: Vysoká škola báňská - Technická univerzita Ostrava, magisterské studium, obor: Aplikovaná mechanika, ukončeno s vyznamenáním  
2008 - dosud: Vysoká škola báňská - Technická univerzita Ostrava, doktorské studium, obor: Aplikovaná mechanika
- Zaměstnání:* 2008 – dosud: Pedagogický a vědeckovýzkumný pracovník, VŠB - Technická univerzita Ostrava, Fakulta strojní, Katedra mechaniky
- Zahraniční stáže:* 2006: Intensive Course of Assistive Technology (Kurs podpůrných technologií pro zdravotně postižené osoby), březen 2006, 14 dní, Jyväskylän University of Applied Science, Finsko  
2006: výměnný pobyt, 4 měsíce, University of Oulu, Finsko;  
2008: výuková stáž, přednášeno 8 hodin
- Ocenění:* 2008: Laureát ceny prof. Babušky za práci v oboru počítačových věd  
2008: Diplomová práce oceněna komisí státních závěrečných zkoušek
- Organizační a vedoucí činnosti:* hlavní i programový vedoucí dvoutýdenního tábora o kapacitě 100 dětí s 15 členým personálem
- Pedagogická praxe:* funkce oddílového vedoucího na dětském táboře v letech 2002 až 2008  
2007 – dosud: Vedení seminářů a přednášek na VŠB – TU Ostrava; i anglicky  
2007 – dosud: Vedoucí 9 bakalářských prací.
- Jazykové znalosti:* angličtina (plynule slovem i písmem), základy ruštiny
- Práce na PC:* operační systémy: Unix/Linux a MS Windows  
aplikace: Matlab, Mathcad, Mathematica, ANSYS, Msc Adams, Pro/ENGINEER, AutoCAD, Inventor, Open Office a MS Office
- Doplňkové kvalifikace:* držitel certifikátu MŠMT opravňujícího vedení dětských táborů a ozdravovacích akcí, řidičský průkaz skupiny B, svářečský průkaz
- Zájmy:* práce s dětmi, dobrovolnictví, šachy (také funkce rozhodčího), cizí jazyky, četba beletrie a odborné literatury, malířské umění období surrealismu a impresionismu, cestování, sport (cyklistika, horolezectví, zimní plavání, košíková, turistika)

## Curriculum Vitae

*First name:* Ondřej

*Surname:* František

*Date of birth:* 2. 8. 1984

*Address:* Gen. Sochora 1204/9, 708 00 Ostrava-Poruba

*Nationality:* Czech Republic

*Marital status:* Single

*Phone, Email:* +420 608 876 862, [ondrej.frantisek@gmail.com](mailto:ondrej.frantisek@gmail.com)

*Education:* 1999 - 2003: Střední průmyslová škola a Střední odborné učiliště Prostějov, Lidická 4,  
majority: Mechanic of Machines and Machinery, concluded with honors

2003 - 2006: VŠB – Technical University of Ostrava, Faculty of Mechanical Engineering,  
bachelor of science with honors, majority: Applied Mechanics

2006 - 2008: VŠB – Technical University of Ostrava, Faculty of Mechanical Engineering,  
master of science with honors, majority: Applied Mechanics

2008 - now: VŠB – Technical University of Ostrava, Faculty of Mechanical  
Engineering, Ph.D. studies, majority: Applied Mechanics

*Job experience:* 2008 - now: VŠB – Technical University of Ostrava, Faculty of Mechanical  
Engineering, Department of Mechanics, position of junior lecturer

*Abroad stays:* 2006: Intensive Course of Assistive Technology (dealing with equipment for disabled  
people), 2 weeks, Jyväskylä University of Applied Science, Finland

2006: exchange student, 4 months, University of Oulu, Finland

2008: lecturing stay, University of Oulu, Finland

*Awards:* 2008: Laureate of prof. Babuška award for thesis in the field of computer science

2008: Award of the leaving exam committee

*Language skills:* English (fluently), basics of Russian

*Computer skills:* operating systems: Unix/Linux (preferred), Windows

software applications: Matlab, Mathcad, Mathematica, ANSYS, Msc Adams,  
Pro/ENGINEER, AutoCAD, Inventor, Open Office a MS Office

*Supplemental skills:* driving license B

*Hobbies:* work with children (voluntary activities), languages, reading of classical and technical  
literature, arts, traveling, sport (rock and mountain climbing, cycling, swimming, tennis)



## 11 Přílohy

### 11.1 Procedury sestavující matice základních pohybů

#### Transformační matice translace ve směru osy x

```
function trans=Tx(x)
%%Vytvoří transformační matici základního pohybu, posuv po x o délku x.
trans=[1 0 0 x; 0 1 0 0; 0 0 1 0; 0 0 0 1];
```

#### Transformační matice translace ve směru osy y

```
function trans=Ty(y)
%%Vytvoří transformační matici základního pohybu, posuv po y o délku y.
trans=[1 0 0 0; 0 1 0 y; 0 0 1 0; 0 0 0 1];
```

#### Transformační matice translace ve směru osy z

```
function trans=Tz(z)
%%Vytvoří transformační matici základního pohybu, posuv po z o délku z.
trans=[1 0 0 0; 0 1 0 0; 0 0 1 z; 0 0 0 1];
```

#### Transformační matice translace do obecného bodu X

```
function trans=T(X)
%%Vytvoří transformační matici základního pohybu, posuv do bodu X.
trans=[1 0 0 X(1); 0 1 0 X(2); 0 0 1 X(3); 0 0 0 1];
```

#### Transformační matice rotace kolem osy x

```
function rotx=Rx(fix)
%%Vytvoří transformační matici základního pohybu, rotace kolem x o úhel fix.
rotx=[1 0 0 0; 0 cos(fix) -sin(fix) 0; 0 sin(fix) cos(fix) 0; 0 0 0 1];
```

#### Transformační matice rotace kolem osy y

```
function roty=Ry(fiy)
%%Vytvoří transformační matici základního pohybu, rotace kolem y o úhel fiy.
roty=[cos(fiy) 0 sin(fiy) 0; 0 1 0 0; -sin(fiy) 0 cos(fiy) 0; 0 0 0 1];
```

#### Transformační matice rotace kolem osy z

```
function rotz=Rz(fiz)
%%Vytvoří transformační matici základního pohybu, rotace kolem z o úhel fiz.
rotz=[cos(fiz) -sin(fiz) 0 0; sin(fiz) cos(fiz) 0 0; 0 0 1 0; 0 0 0 1];
```

## 11.2 Metoda BFGS, modifikovaná BFGS a DFP – numerické testy

V algoritmu II v kroku 7 byly testovány tyto vztahy pro opravy inverzní matice Hessiánu:

Davidon-Fletcher-Powell (dále DFP) z [18]

$$\mathbf{H}_{DFP}^{(k+1)} = \mathbf{H}_{DFP}^{(k)} - \frac{\mathbf{H}_{DFP}^{(k)} \cdot \mathbf{y}^{(k)} \cdot (\mathbf{y}^{(k)})^T \cdot \mathbf{H}_{DFP}^{(k)}}{(\mathbf{y}^{(k)})^T \cdot \mathbf{H}_{DFP}^{(k)} \cdot \mathbf{y}^{(k)}} + \frac{\mathbf{d}^{(k)} \cdot (\mathbf{d}^{(k)})^T}{(\mathbf{y}^{(k)})^T \cdot \mathbf{d}^{(k)}}. \quad (11.1)$$

Broyden-Fletcher-Goldfarb-Shanno (dále BFGS) a modifikace Broyden-Fletcher-Goldfarb-Shanno (dále modifikovaná BFGS) z [23]:

$$\mathbf{H}_{BFGS}^{(k+1)} = \mathbf{H}_{BFGS}^{(k)} + \rho^{(k)} \cdot \left( \left( \lambda^{(k)} + (\mathbf{y}^{(k)})^T \cdot \mathbf{H}_{BFGS}^{(k)} \cdot \mathbf{y}^{(k)} \cdot \rho^{(k)} \right) \cdot \mathbf{d}^{(k)} \cdot (\mathbf{d}^{(k)})^T - \mathbf{d}^{(k)} \cdot (\mathbf{y}^{(k)})^T \cdot \mathbf{H}_{BFGS}^{(k)} - \mathbf{H}_{BFGS}^{(k)} \cdot \mathbf{y}^{(k)} \cdot (\mathbf{d}^{(k)})^T \right),$$

$$\text{kde } \rho^{(k)} = \frac{1}{(\mathbf{d}^{(k)})^T \cdot \mathbf{y}^{(k)}}. \quad (11.2)$$

Broyden-Fletcher-Goldfarb-Shanno:  $\lambda^{(k)} = 1$ .

Modifikace Broyden-Fletcher-Goldfarb-Shanno:  $\lambda^{(k)} = 1/t^{(k)}$ ,

$$\text{kde } t^{(k)} = \frac{6}{(\mathbf{d}^{(k)})^T \cdot \mathbf{y}^{(k)}} \left( F^{(k)} - F^{(k+1)} + (\mathbf{d}^{(k)})^T \cdot \mathbf{g}^{(k+1)} \right) - 2.$$

### 11.2.1 Kombinovaná BFGS metoda

Při hlubším prozkoumání zjistil autor, že nejefektivnější je algoritmus, který na počátku používá k opravám inverzní matice Hessiánu metodu BFGS a později přechází na modifikovanou metodu BFGS.

V praxi bylo pro  $k \leq 9$  použita BFGS matice a pro  $k > 9$  použita modifikovaná BFGS matice.

### 11.2.2 Numerické testy

Aby bylo možné porovnat výše uvedené algoritmy optimalizace, je třeba zvolit funkce, na kterých budou algoritmy testovány a porovnávány. Funkce parabolická je z hlediska matematických metod optimalizace tou nejjednodušší a bude tudíž sloužit pouze k ověření, zda algoritmus funguje. Kvality algoritmů budou prověřovány na Rosenbrockově funkci vykreslené ne obr. 11.1, neboli banánovém údolí, které je jednou z nejčastěji používaných funkcí při testování

optimalizačních algoritmů. Obě funkce budou dvourozměrné, což znamená, že dimenze úlohy je  $n = 2$ .

## Parabolická funkce

Počátečním bodem pro kvazinevtonovské metody bude  $\mathbf{x}^{(0)} = [75; -100]$ . Pro metodu simplexů jsou zapotřebí body tři  $\mathbf{x}_1^{(0)} = [70; -100]$ ,  $\mathbf{x}_2^{(0)} = [80; -100]$ ,  $\mathbf{x}_3^{(0)} = [75; -105]$ . Parabolická funkce má minimum v bodě  $\mathbf{x}_{min} = [0; 0]$  byla zvolena ve tvaru

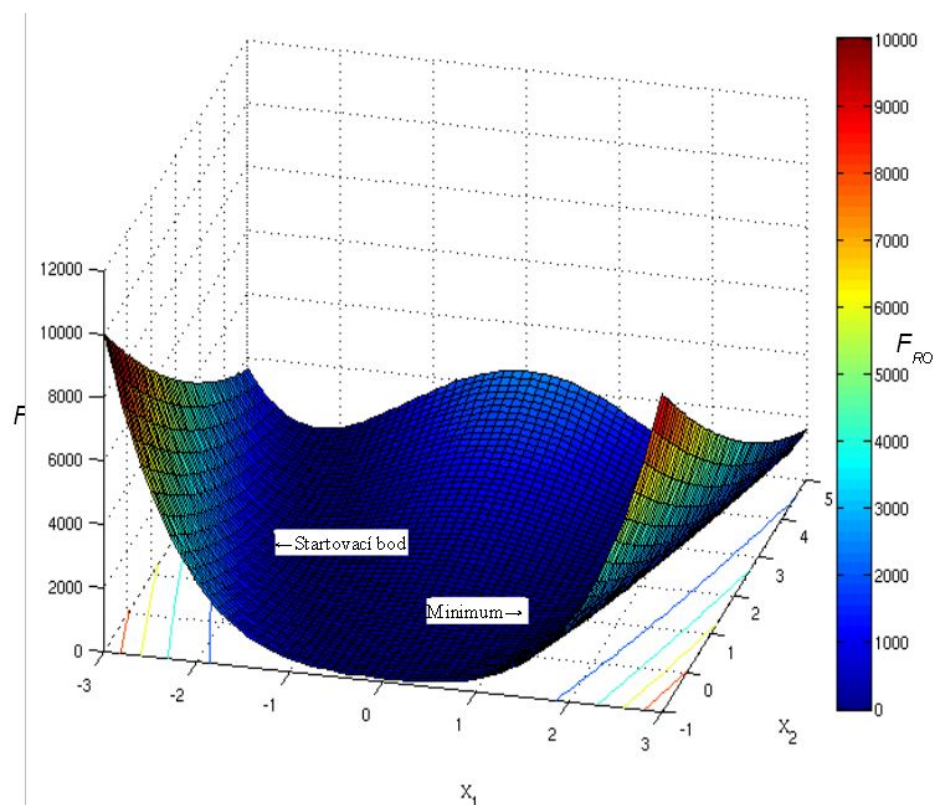
$$F_{par}(\mathbf{x}) = x_1^2 + 7 \cdot x_2^2, \quad F_{par}(\mathbf{x}_{min}) = 0. \quad (11.3)$$

Minimum této cílové funkce bylo nalezeno bez jakýchkoli problémů, a to všemi metodami, tzn. metodou simplexů, metodou DFP, BFGS, modifikovanou BFGS i kombinovanou BFGS.

## Rosenbrockova funkce neboli banánové údolí

Počátečním bodem pro kvazinevtonovské metody bude  $\mathbf{x}^{(0)} = [-1,9; 2]$ . Pro metodu simplexů jsou zapotřebí body tři  $\mathbf{x}_1^{(0)} = [-1,4; 2]$ ,  $\mathbf{x}_2^{(0)} = [-2,4; 2]$ ,  $\mathbf{x}_3^{(0)} = [-1,9; 1,5]$ . Rosenbrockova funkce má minimum v bodě  $\mathbf{x}_{min} = [1; 1]$  a má tvar  $F_{ROS}$

$$F_{Ros}(\mathbf{x}) = 100 \cdot (x_2 - x_1^2)^2 + (1 - x_1)^2, \quad F_{Ros}(\mathbf{x}_{min}) = 0. \quad (11.4)$$



Obrázek 11.1: Třírozměrný graf Rosenbrockovy funkce

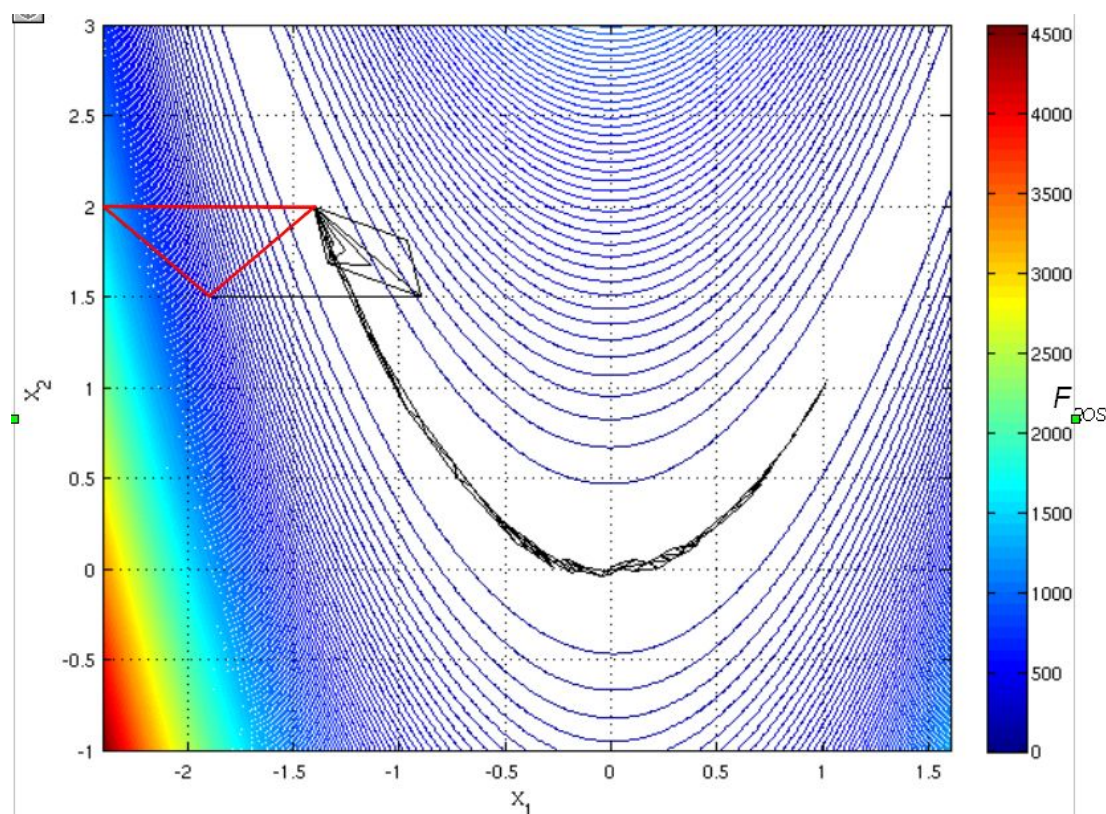
## Numerické srovnání optimalizačních metod

Tabulka 11.1: Numerické srovnání optimalizačních algoritmů

Optimalizační metoda	Počet vyčíslení cílové funkce	Počet iterací	Výsledný vektor ladicích parametrů $x_{min}$
Metoda simplexů ( $\varepsilon = 10^{-4}$ )	276	93	[1,0006; 1,0013]
DFP ( $\varepsilon = 10^{-7}$ )	Minimum nebylo nalezeno!		
BFGS ( $\varepsilon = 10^{-7}$ )	783	260	[1,0000; 1,0000]
Modifikovaná BFGS ( $\varepsilon = 10^{-7}$ )	Minimum nebylo nalezeno!		
Kombinovaná BFGS ( $\varepsilon = 10^{-7}$ )	204	67	[1,0000; 1,0000]

Porovnání algoritmů na banánovém údolí ukázalo zajímavé a překvapující výsledky, které lze vidět v tabulce 11.1. Algoritmy používající matice DFP a modifikovanou BFGS havarovaly a minimum nenalezly, naopak metoda BFGS našla minimum, ale za cenu velkého počtu vyčíslení cílové funkce, takže pro praktické výpočty také není vhodná.

Ve výše uvedených testech se nejlépe osvědčila kombinovaná BFGS metoda a dále metoda simplexů. V jiných případech kombinovaná metoda BFGS havarovala a proto byla pro praktické výpočty využívána stabilní BFGS oprava. Postup metody simplexů při prohledávání terénu Rosenbrockovy funkce je uveden v obr. 11.2.



Obrázek 11.2: Zobrazení simplexů při prohledávání terénu Rosenbrockovy funkce

## 11.3 Algoritmus metody simplexů neboli metody Nelder-Mead

### Hlavní procedura

```
function X=Nelder_Mead(X0,epsNS)
%%Algoritmus optimalizace bez omezení; metoda simplexů neboli Nelder-Mead.
%X-Výsledný vektor ladicích parametrů.
%X0-Počáteční simplex, řádek odpovídá souřadnici vrcholu simplexu.
%epsNS-Hodnota sloužící k ukončení algoritmu.

eps=100;iteraci=1;
s=size(X0);n=s(2);%n-Počet dimenzí úlohy.
F_X=[];%F_X-Matice, kde v prvním sloupci jsou hodnoty cílové funkce a v dalších
%sloupcích jsou odpovídající souřadnice vrcholu simplexu.

for i=1:n+1
    F_X=[F_X;trajektorie_bodu_M(X0(i,:)) X0(i,:)];
end

F_X=order(F_X,n);%Uspořádání vrcholů podle hodnoty cílové funkce.

%Parametry Nelder-Mead metody:
ref=1;%Zrcadlení.
exp=2;%Zvětšení.
con=0.5;%Kontrakce.
shr=0.5;%Zmenšení.

while eps>epsNS
    F_X=order(F_X,n);%Uspořádání vrcholů podle hodnoty cílové funkce.

    xbar=sum(F_X(1:n,2:n+1))*1/n;%Střední hodnota n nejlepších bodů.

    %Výpočet bodu zrcadlení xr:
    xr=(1+ref)*xbar-ref*F_X(n+1,2:n+1);Fr=trajektorie_bodu_M(xr);
    if F_X(1,1)<=Fr && Fr<F_X(n,1); xn=xr; F_X=[F_X(1:n,:);Fr xn]; end

    %Výpočet bodu zvětšení xe:
    if Fr<F_X(1,1)
        xe=(1+ref*exp)*xbar-ref*exp*F_X(n+1,2:n+1);Fe=trajektorie_bodu_M(xe);
        if Fe<Fr; xn=xe; F_X=[F_X(1:n,:);Fe xn];
        else, xn=xr; F_X=[F_X(1:n,:);Fr xn]; end
    end

    %Výpočet kontrakce:
    if Fr>=F_X(n,1)

        %Výpočet bodu vnější kontrakce xc:
        if F_X(n,1)<=Fr && Fr<F_X(n+1,1)
            xc=(1+ref*con)*xbar-ref*con*F_X(n+1,2:n+1);Fc=trajektorie_bodu_M(xc);
            if Fc<=Fr; xn=xc; F_X=[F_X(1:n,:);Fc xn];

            %Výpočet zmenšení:
            else
                F_XS=F_X(1,:);
                for i=2:n+1
                    xs(i,:)=F_X(1,2:n+1)+shr*(F_X(i,2:n+1)-F_X(1,2:n+1));
                F_XS=[F_XS;trajektorie_bodu_M(xs(i,:)) xs(i,:)];
            end

            F_X=F_XS;
        end
    end
end
```

```

        end
    end

    %Výpočet bodu vnitřní kontrakce xcc:
    if Fr>=F_X(n+1,1)
        xcc=(1-con)*xbar+con*F_X(n+1,2:n+1);Fcc=trajektorie_bodu_M(xcc);
        if Fcc<F_X(n+1,1), xn=xcc; F_X=[F_X(1:n,:);Fcc xn];

        %Výpočet zmenšení:
        else
            F_XS=F_X(1,:);
            for i=2:n+1
                xs(i,:)=F_X(1,2:n+1)+shr*(F_X(i,2:n+1)-F_X(1,2:n+1));
                F_XS=[F_XS;trajektorie_bodu_M(xs(i,:)) xs(i,:)];
            end
            F_X=F_XS;
        end
    end

end

end

%Výpočet střední hodnoty všech vrcholů t:
t(iteraci,:)=sum(F_X(1:n+1,2:n+1))*1/(n+1);
if iteraci>1,eps=max(abs(t(iteraci,:)-t(iteraci-1,:)));end

iteraci=iteraci+1,
end

X=sum(F_X(1:n+1,2:n+1))*1/(n+1);iteraci=iteraci-1,

tkon=rem(now,1);beep;
Vyp_cas=tkon-tpoc;disp(['Vypoctovy cas je ',datestr(Vyp_cas,13)]);

```

## Procedura sloužící k uspořádání vrcholů simplexů

```

function A=order(M,n),
%%Algoritmus uspořádá řádky matice M podle velikosti prvků v prvním sloupci.
%A-Výsledná uspořádaná matice.
%M-Matice určená pro uspořádání.
%n-Dimenze optimalizační úlohy.

F=sort(M(:,1));A=[];
for i=1:n+1,
    for j=1:n+1,
        if M(j,1)==F(i,1)
            A(i,:)=M(j,:);
        end
    end
end
end

```

## 11.4 Algoritmus metody BFGS

```
function x=BFGS(X0,epsBFGS)
%%Algoritmus optimalizace bez omezení; kombinovaná metoda BFGS
%x-Výsledný vektor ladicích parametrů.
%X0-Počáteční řádkový vektor ladicích parametrů.
%epsBFGS-Hodnota sloužící k ukončení algoritmu.

epsilon=100;tpoc=rem(now,1);iteraci=0;
h=sqrt(eps);%h-Hodnota definující okolí, na kterém se určuje numerická derivace.

xc=X0;
s=size(X0);n=s(2);%Počet dimenzí úlohy.
E=eye(n);%E-Matice identity

%Výpočet gradientu gc v bodě xc:
fc=trajektorie_bodu_M(xc),%Vyčíslení hodnoty cílové funkce v bodě xc
for i=1:n
    gc(i)=(trajektorie_bodu_M(xc+h*E(i,:))-fc)/h;
end

%Nástřel počátečního inverzního Hessiánu:
p=1;Hc=p*E;

while epsNS<epsilon;
    iteraci=1+iteraci,

    dc=-Hc*gc';%d-Krok; sloupcový vektor.

    xn=xc+dc';%xn-Nový bod.

    fn=trajektorie_bodu_M(xn),%Vyčíslení hodnoty cílové funkce v novém bodě xn.

    %Výpočet gradientu gc v bodě xc:
    for i=1:n
        gn(i)=(trajektorie_bodu_M(xn+h*E(i,:))-fn)/h;
    end

    yc=(gn-gc)';%Výpočet rozdílu gradientů v xn a xc.

    lam=1;

    %Aktualizace Hessiánu BFGS
    Hn=Hc+1/(dc'*yc)*( (lam+yc'*Hc*yc/(dc'*yc)) *dc*dc'-dc*yc'*Hc-Hc*yc*dc' );

    if iteraci>1
        epsilon=max(abs(xc-xn));
    end

    Hc=Hn;gc=gn;xc=xn,fc=fn;

end

beep;x=xc;iteraci,tkon=rem(now,1);Vyp_cas=tkon-tpoc,
disp(['Vypoctovy cas je ',datestr(Vyp_cas,13)]);
```

## 11.5 Metoda penalizace

### Hlavní procedura

```
function [x,c,Fmin,konv]=penalty_function(x0,beta,mi0,v,maxit,epsBFGS)
%Metoda resí problémy optimalizace s omezovacími podmínkami
%x - Výsledný vektor ladicích parametrů.
%c - vektor velikosti porušení jednotlivých podmínek
%Fmin - hodnota cílové funkce pro xmin
%x0 - Počáteční řádkový vektor ladicích parametrů.
%v - tolerance porušení podmínek omezení
%faktor >1 - násobné zvyšování penalizačního parametru
% mi0 - počáteční penalizační parametr >0, doporučen malý např. 0.1
% maxit - maximální počet iterací
% i - poradové číslo iterace
%epsBFGS-Hodnota sloužící k ukončení algoritmu BFGS (optimalizace bez omezení).
% konv=0 - úloha konvergovala
% konv=1 - úloha nekonvergovala

global mi
mi=mi0;x=x0;i=1;

x=BFGS(x,epsBFGS);%hledá minimum funkce Qpen

while max(constrain(x))>=v
    i=i+1;
    if i>maxit
        display('nekonverguje')
        konv=1;
        break
    end
    mi=beta*mi;
    x=BFGS(x,epsBFGS);%hledá minimum funkce Qpen
    konv=0;
end
c=constrain(x);Fmin=Fcost(x);
```

### Procedura sestavující penalizovanou funkci $Q(x, \mu)$

```
function res=Qpen(x)
%Vypočte hodnotu penalizované cílové funkce

global mi
res=Fcost(x)+mi*constraints(x)'*constraints(x);
```

### Procedura definice podmínek omezení<sup>20</sup>

```
function c=constraints(x)
%Definice podmínek omezení
b=60;vule=35;L=332;

%Rovnostní omezující podmínky
crov=[];

%Nerovnostní omezující podmínky ve tvaru c(i)<=0
cnerov=[-(L4-x(1)+x(2))/2+(b+vule)
        -x(1)+210
```

<sup>20</sup> Konkrétní omezující podmínky uvedené v proceduře odpovídají podmínkám definovaným v 7.42.



```

x(2)-100];
n=size(c);
cnerov=max(zeros(n(1),1),cnerov);%0setruje nerovnost

%Sloceni rovostnich a nerovnostnich podminek

c=[crov;cnerov];

```

## 11.6 Algoritmus Newtonovy metody řešení soustav nelineárních rovnic

```

function [x0,konv]=newton_algorithm(x0,itmax,epsNS)
%x0 - počáteční odhad řešení soustavy
%itmax - maximální počet iterací
%epsNS - limitní normu reziduálů sloužící pro ukončení výpočtu
%i - pořadové číslo iterace
% funkce f nonlin_system vyjadřuje soustavu nelineárních rovnic a musí být
% ve tvaru, aby pro řešení byla f=0
% konv=0 - úloha konvergovala
% konv=1 - úloha nekonvergovala

i=1;eps=100;

while epsNS<eps
    F0=nonlin_system(x0);
    J=numJacobian(x0,F0);
    p=J\(-F0);
    x0=x0+p;
    i=i+1;
    eps=norm(F0);
    if i==itmax
        display('nekonverguje');
        konv=1;
        break
    end
    konv=0;
end
end

function J=numJacobian(x0,F0)
%vnořená funkce - vypočte Jacobián soustavy rovnic pomocí numerické
%derivace
n=max(size(x0));E=eye(n);h=sqrt(eps);J=[];
for i=1:n
    J=[J (nonlin_system(x0+h*E(:,i))-F0)*1/h];
end
end

```

## 11.7 Algoritmus Broydenovy dobré metody řešení soustav nelineárních rovnic

```

function [x0,konv]=broyden_algorithm(x0,itmax,epsNS)
%x0 - počáteční odhad řešení soustavy
%itmax - maximální počet iterací
%epsNS - limitní normu reziduálů sloužící pro ukončení výpočtu
%i - pořadové číslo iterace

```

---

```

% funkce f nonlin_system vyjadřuje soustavu nelineárních rovnic a musí být
% ve tvaru, aby pro řešení byla f=0
% konv=0 - úloha konvergovala
% konv=1 - úloha nekonvergovala

i=1;eps=100;n=max(size(x0));      F0=nonlin_system(x0);

J0=numJacobian(x0,F0);H0=inv(J0);

while epsNS<eps
    p=H0*(-F0);
    x1=x0+1*p;
    F1=nonlin_system(x1);
    y0=F1-F0;
    H1=H0+((p-H0*y0)*p'*H0)/(p'*H0*y0);
    x0=x1;F0=F1;H0=H1;
    i=i+1;
    eps=norm(F0);
    if i==itmax
        display('nekonverguje')
        konv=1;
        break
    end
    konv=0;
end
end

function J=numJacobian(x0,F0)
%vnořená funkce - vypočte Jacobián soustavy rovnic pomocí numerické
%derivace - pouze na počátku výpočtu - pro i=1
n=max(size(x0));E=eye(n);h=sqrt(eps);J=[];
for i=1:n
    J=[J (nonlin_system(x0+h*E(:,i))-F0)*1/h];
end
end

```

## 11.8 Algoritmus Martinezovy hybridní metody řešení soustav nelineárních rovnic

```

function x0=martinez_hybrid_method(x0,itmax,epsNS)
%x0 - počáteční odhad řešení soustavy
%itmax - maximální počet iterací
%epsNS - limitní normu reziduálů sloužící pro ukončení výpočtu
%i - pořadové číslo iterace
% funkce f nonlin_system vyjadřuje soustavu nelineárních rovnic a musí být
% ve tvaru, aby pro řešení byla f=0
% p0 - krok
% tol - tolerance podmínky nesestavitelnosti doporučeno tol>10
% konv=0 - úloha konvergovala
% konv=1 - úloha nekonvergovala
% konv=2 - úloha nekonvergovala - nesestavitelný mechanismus

tol=100;i=1;eps=100;F0=nonlin_system(x0);

J0=numJacobian(x0,F0);H0=inv(J0);

while epsNS<eps

    pk=H0*(-F0);

```

```

if i==1
    ppoc=pk;
end

if pk>tol*ppoc
    display('nekonverguje-nesetavitelny mechanismus')
    konv=2;
    break
end

x1=x0+pk;
F1=nonlin_system(x1);
y0=F1-F0;

if i>1 && abs(pk'*pmin1)/(pk'*H0*y0)>=abs(y0'*ymin1)/abs(y0'*y0)
    H1=H0+(pk-H0*y0)*y0'/(y0'*y0);
else
    H1=H0+((pk-H0*y0)*pk'*H0)/(pk'*H0*y0);
end

x0=x1;F0=F1;H0=H1;pmin1=pk;ymin1=y0;
i=i+1;
eps=norm(F0);

if i==itmax
    display('nekonverguje')
    konv=1;
    break
end
konv=0;
end
end

function J=numJacobian(x0,F0)
%vnořená funkce - vypočte Jacobián soustavy rovnic pomocí numerické
%derivace - pouze na počátku výpočtu - pro i=1
n=max(size(x0));E=eye(n);h=sqrt(eps);J=[];
for i=1:n
    J=[J (nonlin_system(x0+h*E(:,i))-F0)*1/h];
end
end

```

## 11.10 Úloha polohy levostranného mechanismu řazení

### 11.10.1 Hlavní procedura

```
function F=trajektorie_bodu_M%uprava1
%%Vykreslí trajektorii bodu M rukojeti řadicí páky 8 a stanoví hodnotu cílové
%funkce F.

i=1;clear global

itmax=100;epsNS=1e-11;

n=15;%Počet bodů, v kterých je počítána jedna část trajektorie bodu M. Například
%cesta z neutrálu k druhému rychlostnímu stupni.

pmax=18;fiimax=pi/180*(8);%Krajní hodnoty vstupních souřadnic.

%uprava2

rozmary;%Načte rozměry mechanismu.

global p fii xS1 yS1 zS1 L8p q_poc_poloha;p=0;fii=0;

%Vyřeší zdvihovou závislost pro neutrálu:
q=martinez_hybrid_method(q_poc_poloha,itmax,epsNS);qN=q;
T81=Rx(q(16))*Ry(q(17))*Rz(q(18))*Tx(-xS1)*Ty(-yS1)*Tz(-zS1);
rM1=inv(T81)*[0 0 L8p 1]';M1(i,:)=rM1(1:2,1)';

%Cyklus řeší cestu bodu M při řazení třetího rychlostního stupně z neutrálu:
for j=1:n
    q_poc_poloha=q; p=j/n*pmax; i=i+1;
    q=martinez_hybrid_method(q_poc_poloha,itmax,epsNS);
    T81=Rx(q(16))*Ry(q(17))*Rz(q(18))*Tx(-xS1)*Ty(-yS1)*Tz(-zS1);
    rM1=inv(T81)*[0 0 L8p 1]';M1(i,:)=rM1(1:2,1)';
end
treti_rychl_stupen=M1;%Matice souřadnic bodu M; v řádcích xM1,yM1.

%Cyklus řeší cestu bodu M při řazení druhého rychlostního stupně z neutrálu:
i=1;
q=qN;
for j=1:n
    q_poc_poloha=q; p=-j/n*pmax; i=i+1;
    q=martinez_hybrid_method(q_poc_poloha,itmax,epsNS);
    T81=Rx(q(16))*Ry(q(17))*Rz(q(18))*Tx(-xS1)*Ty(-yS1)*Tz(-zS1);
    rM1=inv(T81)*[0 0 L8p 1]';M1(i,:)=rM1(1:2,1)';
end
druhy_rychl_stupen=M1;%Matice souřadnic bodu M; v řádcích xM1,yM1.

%Cyklus řeší pohyb bodu M zvaný volba směrem k bodu N45 tudíž ke čtvrtému a
%pátému rychlostnímu stupni z neutrálu:
i=1;p=0;
q=qN;
```

```

for j=1:n
    q_poc_poloha=q; fii=-j/n*fiimax; i=i+1;
    q=martinez_hybrid_method(q_poc_poloha,itmax,epsNS);
    T81=Rx(q(16))*Ry(q(17))*Rz(q(18))*Tx(-xS1)*Ty(-yS1)*Tz(-zS1);
    rM1=inv(T81)*[0 0 L8p 1]';M1(i,:)=rM1(1:2,1)';
end
volba_4_5=M1;%Matice souřadnic bodu M; v řádcích xM1,yM1.

%Cyklus řeší cestu bodu M při řazení čtvrtého rychlostního stupně z bodu N45:
M1=[];i=0;
q45=q;
for j=1:n
    q_poc_poloha=q; p=-j/n*pmax; i=i+1;
    q=martinez_hybrid_method(q_poc_poloha,itmax,epsNS);
    T81=Rx(q(16))*Ry(q(17))*Rz(q(18))*Tx(-xS1)*Ty(-yS1)*Tz(-zS1);
    rM1=inv(T81)*[0 0 L8p 1]';M1(i,:)=rM1(1:2,1)';
end
ctvrty_rychl_stupen=[volba_4_5(end,:);M1];%Matice souřadnic bodu M; v řádcích
%xM1,yM1.

%Cyklus řeší cestu bodu M při řazení pátého rychlostního stupně z bodu N45:
i=0;M1=[];
q=q45;
for j=1:n
    q_poc_poloha=q; p=j/n*pmax; i=i+1;
    q=martinez_hybrid_method(q_poc_poloha,itmax,epsNS);
    T81=Rx(q(16))*Ry(q(17))*Rz(q(18))*Tx(-xS1)*Ty(-yS1)*Tz(-zS1);
    rM1=inv(T81)*[0 0 L8p 1]';M1(i,:)=rM1(1:2,1)';
end
paty_rychl_stupen=[volba_4_5(end,:);M1];%Matice souřadnic bodu M; v řádcích
%xM1,yM1.

%Cyklus řeší pohyb bodu M zvaný volba směrem k bodu NR1 tudíž ke zpátečce a
%prvnímu rychlostnímu stupni z neutrálu:
i=0;p=0;M1=[];
q=qN;
for j=1:n
    q_poc_poloha=q; fii=j/n*fiimax; i=i+1;
    q=martinez_hybrid_method(q_poc_poloha,itmax,epsNS);
    T81=Rx(q(16))*Ry(q(17))*Rz(q(18))*Tx(-xS1)*Ty(-yS1)*Tz(-zS1);
    rM1=inv(T81)*[0 0 L8p 1]';M1(i,:)=rM1(1:2,1)';
end
volba_R_1=[tretí_rychl_stupen(1,:);M1];%Matice souřadnic bodu M; v řádcích
%xM1,yM1.

%Cyklus řeší cestu bodu M při řazení zpátečky z bodu N45:
M1=[];i=0;
qR1=q;
for j=1:n
    q_poc_poloha=q; p=-j/n*pmax; i=i+1;
    q=martinez_hybrid_method(q_poc_poloha,itmax,epsNS);
    T81=Rx(q(16))*Ry(q(17))*Rz(q(18))*Tx(-xS1)*Ty(-yS1)*Tz(-zS1);
    rM1=inv(T81)*[0 0 L8p 1]';M1(i,:)=rM1(1:2,1)';
end
zpatecka=[volba_R_1(end,:);M1];%Matice souřadnic bodu M; v řádcích xM1,yM1.

%Cyklus řeší cestu bodu M při řazení prvního rychlostního stupně z bodu NR1:
i=0;M1=[];
q=qR1;
for j=1:n
    q_poc_poloha=q; p=j/n*pmax; i=i+1;

```

```

q=martinez_hybrid_method(q_poc_poloha,itmax,epsNS);
T81=Rx(q(16))*Ry(q(17))*Rz(q(18))*Tx(-xS1)*Ty(-yS1)*Tz(-zS1);
rM1=inv(T81)*[0 0 L8p 1]';M1(i,:)=rM1(1:2,1)';
end
prvni_rychl_stupen=[volba_R_1(end,:);M1];%Matice souřadnic bodu M; v řádcích
%xM1,yM1.

%Vykreslování trajektorií bodu M:
%uprava3zacatek
h=figure;
axes('Parent',h,'XDir','reverse','FontWeight','light','FontSize',16);
hold on;grid on;axis equal;
plot(volba_R_1(:,2),volba_R_1(:,1),'linewidth',3);
plot(zpatecka(:,2),zpatecka(:,1),'linewidth',3);
plot(prvni_rychl_stupen(:,2),prvni_rychl_stupen(:,1),'linewidth',3);
plot(druhy_rychl_stupen(:,2),druhy_rychl_stupen(:,1),'linewidth',3);
plot(treti_rychl_stupen(:,2),treti_rychl_stupen(:,1),'linewidth',3);
plot(volba_4_5(:,2),volba_4_5(:,1),'linewidth',3);
plot(ctvrty_rychl_stupen(:,2),ctvrty_rychl_stupen(:,1),'linewidth',3);
plot(paty_rychl_stupen(:,2),paty_rychl_stupen(:,1),'linewidth',3);
xlabel('y_1 [mm] ','fontangle','italic','FontSize',20)
ylabel('x_1 [mm] ','fontangle','italic','FontSize',20)
title('{Trajektorie bodu M v globálním souřadném systému \it\fontsize{19}...
{y_1-x_1}}','FontSize',20)
%uprava3konec

%%Výpočet cílové funkce:
ws=0.5;wd=0.5;%váhové faktory.
%Funkce trajektorie z prvního rychlostního stupně ke zpátečce:
x1=prvni_rychl_stupen(end,1);y1=prvni_rychl_stupen(end,2);
xR=zpatecka(end,1);yR=zpatecka(end,2);
a=yR-y1;b=x1-xR;c=xR*(y1-yR)-yR*(x1-xR);
x1R=volba_R_1(end,1);y1R=volba_R_1(end,2);
d=(a*x1R+b*y1R+c)/sqrt(a^2+b^2);
sklon=yR-y1;
GR1=(ws*sklon)^2+(wd*d)^2;

%Funkce trajektorie z druhého do třetího rychlostního stupně:
x2=druhy_rychl_stupen(end,1);y2=druhy_rychl_stupen(end,2);
x3=treti_rychl_stupen(end,1);y3=treti_rychl_stupen(end,2);
a=y2-y3;b=x3-x2;c=x2*(y3-y2)-y2*(x3-x2);
xN=druhy_rychl_stupen(1,1);yN=druhy_rychl_stupen(1,2);
d=(a*xN+b*yN+c)/sqrt(a^2+b^2);
sklon=y2-y3;
G23=(ws*sklon)^2+(wd*d)^2;

%Funkce trajektorie z čtvrtého do pátého rychlostního stupně:
x4=ctvrty_rychl_stupen(end,1);y4=ctvrty_rychl_stupen(end,2);
x5=paty_rychl_stupen(end,1);y5=paty_rychl_stupen(end,2);
a=y4-y5;b=x5-x4;c=x4*(y5-y4)-y4*(x5-x4);
x45=volba_4_5(end,1);y45=volba_4_5(end,2);
d=(a*x45+b*y45+c)/sqrt(a^2+b^2);
sklon=y4-y5;
G45=(ws*sklon)^2+(wd*d)^2;

Ld=xR-x1;%Délka drah řazení.

%Cílová funkce:
F=(GR1+G23+G45)/Ld^2;

```

### 11.10.2 Procedura definující rozměry levostranného mechanismu

```
function rozmary
%%Definuje rozměry mechanismu.

%Délky jednotlivých členů mechanismu:
global L3k L4 L4s L4v L5d L5h L6r L6s L6v L6h L7 L8r L8s L8p %uprava4
L3k=16;L4=332;L4s=252;%uprava4
L4v=60;L5d=395;L5h=395;L6r=213;L6s=82;
L6v=63;L6h=121;L8r=28;L8s=60;L8p=410;
L7=sqrt((L6s-L4+L4s)^2+(L6v+L4v)^2+(L5d+L5h)^2);

%Souřadnice bodu R a S:
global xR1 yR1 zR1 xS1 yS1 zS1
xR1=348;yR1=0;zR1=395;xS1=589;yS1=0;zS1=851;

%Definuje počáteční polohu mechanismu:
global q_poc_poloha
%Odhad vektoru q v počáteční poloze, tzn. p=0, fii=0:
q_poc_poloha=[ 0 %q(1)
               0 %q(2)
               0 %q(3)
               0 %q(4)
               0 %q(5)
               0 %q(6)
               0 %q(7)
               0 %q(8)
               0 %q(9)
               pi/180*(-5) %q(10)
               pi/180*(5) %q(11)
               0 %q(12)
               pi/180*(5) %q(13)
               pi/180*(-5) %q(14)
               0 %q(15)
               0 %q(16)
               0 %q(17)
               0]; %q(18)
```

### 11.10.3 Procedura sestavující vektor hodnot reziduí algebraických rovnic získaných ze základních smyček

```
function Rez=nonlin_system(q)
%Sestaví vektor Rez hodnot reziduí algebraických rovnic získaných ze základních
%smyček; aby byla splněna úlohy polohy, měl by být vektor nulový tudíž všechna
%rezidua rovny nule.

Rez=[zakladni_smycka_A(q); zakladni_smycka_B(q); zakladni_smycka_C(q)];
```

### 11.10.4 Procedura základní smyčky A

```
function vektor_rc=zakladni_smycka_A(q)
%Sestaví maticovou rovnici základní smyčky A; vybere rezidua šesti nezávislých
%rovníc, které by měli být rovny nule, a ty seřadí do výsledného vektoru
%vektor_rc.

global p fii L3k L4 L5d xR1 yR1 zR1

T14=Tx(p)*Rx(fii)*Ry(q(1))*Tx(L3k)*Rz(q(2));
```

```
T45=Tx(L4)*Rx(q(3))*Ry(q(4))*Rz(q(5))*Tz(L5d);
T51=Ry(q(6))*Tx(-xR1)*Ty(-yR1)*Tz(-zR1);

A=T14*T45*T51-eye(4);

vektor_rc=[A(1,2);A(1,3);A(1,4);A(2,3);A(2,4);A(3,4)];
```

### 11.10.5 Procedura základní smyčky B

```
function vektor_rc=zakladni_smycka_B(q)
%Sestaví maticovou rovnici základní smyčky B; vybere rezidua šesti nezávislých
%rovníc, které by měli být rovny nule, a ty seřadí do výsledného vektoru
%vektor_rc.

global L4 L5d L5h L6s L6v L7 L4v L4s

T45=Tx(L4)*Rx(q(3))*Ry(q(4))*Rz(q(5))*Tz(L5d);
T56=Tz(L5h)*Rx(q(7))*Ry(q(8))*Rz(q(9));
T67=Tx(-L6s)*Ty(L6v)*Rx(q(10))*Ry(q(11))*Rz(q(12));
T74=Tz(-L7)*Rx(q(13))*Ry(q(14))*Ty(L4v)*Tx(-L4s);

B=T45*T56*T67*T74-eye(4);

vektor_rc=[B(1,2);B(1,3);B(1,4);B(2,3);B(2,4);B(3,4)];
```

### 11.10.6 Procedura základní smyčky C

```
function vektor_rc=zakladni_smycka_C(q)
%Sestaví maticovou rovnici základní smyčky C; vybere rezidua šesti nezávislých
%rovníc, které by měli být rovny nule, a ty seřadí do výsledného vektoru
%vektor_rc.

global xR1 yR1 zR1 L5h L6r L6h L8r L8s xS1 yS1 zS1

T15=Tx(xR1)*Ty(yR1)*Tz(zR1)*Ry(-q(6));
T56=Tz(L5h)*Rx(q(7))*Ry(q(8))*Rz(q(9));
T68=Tx(L6r)*Tz(L6h)*Ry(q(15))*Tx(L8r)*Tz(-L8s);
T81=Rx(q(16))*Ry(q(17))*Rz(q(18))*Tx(-xS1)*Ty(-yS1)*Tz(-zS1);

C=T15*T56*T68*T81-eye(4);

vektor_rc=[C(1,2);C(1,3);C(1,4);C(2,3);C(2,4);C(3,4)];
```

### 11.10.7 Optimalizace levostranného mechanismu

Pokud bychom chtěli optimalizovat levostranný mechanismus řazení, lze použít výše uvedené procedury, ve kterých je třeba udělat následující změny. Každá změna bude odkázána na proceduru, ve které má být provedena, a na místo, kde je třeba algoritmus upravit. Místo úpravy je označeno číslem, například `%uprava1`. Jako hlavní proceduru řídící samotný proces optimalizace lze beze změn zvolit algoritmus uvedený v příloze 11.4 nebo 11.3. Postup je uveden pro dvourozměrnou optimalizaci, jejíž výsledky jsou prezentovány v kap. 7.9.1 a 7.9.2.



### **Změna v proceduře 11.10.1**

`%uprava1` – vložit: `(ladici_parametry)`

`%uprava2` – vložit: `global L4s L6s`

`L4s=ladici_parametry(1);L6s=ladici_parametry(2);`

`%uprava3zacatek` - `%uprava3konec` – vymazat řádky mezi značkami

### **Změna v proceduře 11.10.2**

`%uprava4` – smazat ladící parametry, které budou optimalizovány

## 11.11 Úloha polohy pravostranného mechanismu řazení

### 11.11.1 Hlavní procedura

```
function F=trajektorie_bodu_M
%Vykreslí trajektorii bodu M rukojeti řadící páky 12 a stanoví hodnotu cílové
%funkce F.

i=1;clear global;itmax=100;epsNS=1e-11;

n=3;%Počet bodů, v kterých je počítána jedna trajektorie boudy M. Například z
%neutrálu N34 ke třetímu rychlostnímu stupni.

pmax=18;fiimax=pi/180*(8);%Krajní hodnoty vstupních souřadnic.

rozmary%Načte rozměry mechanismu.

global p fii RS1 M12 q_poc_poloha;p=0;fii=0;

%Vyřeší zdvihovou závislost pro neutrál N34:
q=martinez_hybrid_method(q_poc_poloha,itmax,epsNS);qN34=q;
T12_1=Rx(q(29))*Ry(q(30))*Rz(q(31))*T(-RS1);
rM1=inv(T12_1)*[M12 1]';M1(i,:)=rM1(1:2,1)';%Matice souřadnic bodu M; v řádcích
%xM1,yM1.

%Cyklus řeší cestu bodu M při řazení čtvrtého rychlostního stupně z N34:
for j=1:n
    q_poc_poloha=q; p=j/n*pmax; i=i+1;
    q=martinez_hybrid_method(q_poc_poloha,itmax,epsNS);
    T12_1=Rx(q(29))*Ry(q(30))*Rz(q(31))*T(-RS1);
    rM1=inv(T12_1)*[M12 1]';M1(i,:)=rM1(1:2,1)';
end
ctvrty_rychl_stupen=M1;%Matice souřadnic bodu M; v řádcích xM1,yM1.

%Cyklus řeší cestu bodu M při řazení třetího rychlostního stupně z N34:
i=1;
q=qN34;
for j=1:n
    q_poc_poloha=q; p=-j/n*pmax; i=i+1;
    q=martinez_hybrid_method(q_poc_poloha,itmax,epsNS);
    T12_1=Rx(q(29))*Ry(q(30))*Rz(q(31))*T(-RS1);
    rM1=inv(T12_1)*[M12 1]';M1(i,:)=rM1(1:2,1)';
end
treti_rychl_stupen=M1;%Matice souřadnic bodu M; v řádcích xM1,yM1.

%Cyklus řeší pohyb bodu M zvaný volba směrem ke N56 z N34:
i=1;p=0;
q=qN34;
for j=1:n
    q_poc_poloha=q; fii=-j/n*fiimax; i=i+1;
    q=martinez_hybrid_method(q_poc_poloha,itmax,epsNS);
    T12_1=Rx(q(29))*Ry(q(30))*Rz(q(31))*T(-RS1);
    rM1=inv(T12_1)*[M12 1]';M1(i,:)=rM1(1:2,1)';
end
volba_N34_N56=M1;%Matice souřadnic bodu M; v řádcích xM1,yM1.
```

```
%Cyklus řeší cestu bodu M při řazení pátého rychlostního stupně z N56:
M1=[];i=0;
q56=q;
for j=1:n
    q_poc_poloha=q; p=-j/n*pmax; i=i+1;
    q=martinez_hybrid_method(q_poc_poloha,itmax,epsNS);
    T12_1=Rx(q(29))*Ry(q(30))*Rz(q(31))*T(-RS1);
    rM1=inv(T12_1)*[M12 1]';M1(i,:)=rM1(1:2,1)';
end
paty_rychl_stupen=[volba_N34_N56(end,:);M1];%Matice souřadnic bodu M; v řádcích
%xM1,yM1.

%Cyklus řeší cestu bodu M při řazení šestého rychlostního stupně z N56:
i=0;M1=[];
q=q56;
for j=1:n
    q_poc_poloha=q; p=j/n*pmax; i=i+1;
    q=martinez_hybrid_method(q_poc_poloha,itmax,epsNS);
    T12_1=Rx(q(29))*Ry(q(30))*Rz(q(31))*T(-RS1);
    rM1=inv(T12_1)*[M12 1]';M1(i,:)=rM1(1:2,1)';
end
sestý_rychl_stupen=[volba_N34_N56(end,:);M1];%Matice souřadnic bodu M; v řádcích
%xM1,yM1.

%Cyklus řeší pohyb bodu M zvaný volba směrem k bodu N12 z N34:
i=0;p=0;M1=[];
q=qN34;
for j=1:n
    q_poc_poloha=q; fii=j/n*fiimax; i=i+1;
    q=martinez_hybrid_method(q_poc_poloha,itmax,epsNS);
    T12_1=Rx(q(29))*Ry(q(30))*Rz(q(31))*T(-RS1);
    rM1=inv(T12_1)*[M12 1]';M1(i,:)=rM1(1:2,1)';
end
volba_N34_N12=[ctvrty_rychl_stupen(1,:);M1];%Matice souřadnic bodu M; v řádcích
%xM1,yM1.

%Cyklus řeší cestu bodu M při řazení prvního rychlostního stupně z N12:
M1=[];i=0;
qN12=q;
for j=1:n
    q_poc_poloha=q; p=-j/n*pmax; i=i+1;
    q=martinez_hybrid_method(q_poc_poloha,itmax,epsNS);
    T12_1=Rx(q(29))*Ry(q(30))*Rz(q(31))*T(-RS1);
    rM1=inv(T12_1)*[M12 1]';M1(i,:)=rM1(1:2,1)';
end
první_rychl_stupen=[volba_N34_N12(end,:);M1];%Matice souřadnic bodu M; v řádcích
%xM1,yM1.

%Cyklus řeší cestu bodu M při řazení druhého rychlostního stupně z N12:
i=0;M1=[];
q=qN12;
for j=1:n
    q_poc_poloha=q; p=j/n*pmax; i=i+1;
    q=martinez_hybrid_method(q_poc_poloha,itmax,epsNS);
    T12_1=Rx(q(29))*Ry(q(30))*Rz(q(31))*T(-RS1);
    rM1=inv(T12_1)*[M12 1]';M1(i,:)=rM1(1:2,1)';
end
druhý_rychl_stupen=[volba_N34_N12(end,:);M1];%Matice souřadnic bodu M; v řádcích
%xM1,yM1.
```

```

%Cyklus řeší pohyb bodu M zvaný volba směrem k bodu NRC z N12:
i=0;p=0;M1=[];
q=qN12;
for j=1:n
    q_poc_poloha=q; fii=fiimax+j/n*fiimax; i=i+1;
    q=martinez_hybrid_method(q_poc_poloha,itmax,epsNS);
    T12_1=Rx(q(29))*Ry(q(30))*Rz(q(31))*T(-RS1);
    rM1=inv(T12_1)*[M12 1]';M1(i,:)=rM1(1:2,1)';
end
volba_N12_NRC=[volba_N34_N12(end,:);M1];

%Cyklus řeší cestu bodu M při řazení zpátečky z NRC:
M1=[];i=0;
q=qN34;
for j=1:n
    q_poc_poloha=q; p=-j/n*pmax; i=i+1;
    q=martinez_hybrid_method(q_poc_poloha,itmax,epsNS);
    T12_1=Rx(q(29))*Ry(q(30))*Rz(q(31))*T(-RS1);
    rM1=inv(T12_1)*[M12 1]';M1(i,:)=rM1(1:2,1)';
end
zpatecka=[volba_N12_NRC(end,:);M1];%Matice souřadnic bodu M; v řádcích xM1,yM1.

%Cyklus řeší cestu bodu M při řazení crawlingu z NRC:
i=0;M1=[];
q=qN34;
for j=1:n
    q_poc_poloha=q; p=j/n*pmax; i=i+1;
    q=martinez_hybrid_method(q_poc_poloha,itmax,epsNS);
    T12_1=Rx(q(29))*Ry(q(30))*Rz(q(31))*T(-RS1);
    rM1=inv(T12_1)*[M12 1]';M1(i,:)=rM1(1:2,1)';
end
crawling=[volba_N12_NRC(end,:);M1];%Matice souřadnic bodu M; v řádcích xM1,yM1.

%Vykreslování trajektorií bodu M:
h=figure;
axes('Parent',h,'FontWeight','light','FontSize',16);
hold on;grid on;
plot(treti_rychl_stupen(:,1),treti_rychl_stupen(:,2),'linewidth',3);
plot(ctvrty_rychl_stupen(:,1),ctvrty_rychl_stupen(:,2),'linewidth',3);
plot(volba_N34_N56(:,1),volba_N34_N56(:,2),'linewidth',3);
plot(paty_rychl_stupen(:,1),paty_rychl_stupen(:,2),'linewidth',3);
plot(sesty_rychl_stupen(:,1),sesty_rychl_stupen(:,2),'linewidth',3);
plot(volba_N34_N12(:,1),volba_N34_N12(:,2),'linewidth',3);
plot(prvni_rychl_stupen(:,1),prvni_rychl_stupen(:,2),'linewidth',3);
plot(druhy_rychl_stupen(:,1),druhy_rychl_stupen(:,2),'linewidth',3);
plot(volba_N12_NRC(:,1),volba_N12_NRC(:,2),'linewidth',3);
plot(zpatecka(:,1),zpatecka(:,2),'linewidth',3);
plot(crawling(:,1),crawling(:,2),'linewidth',3);
xlabel('x_1 [mm] ','fontangle','italic','FontSize',20)
ylabel('y_1 [mm] ','fontangle','italic','FontSize',20)
title('{Trajektorie bodu M v globálním souřadném systému \it\fontsize{19}...
{y_1-x_1}}','FontSize',20)

%%Výpočet cílové funkce:
ws=0.5;wd=0.5;%váhové faktory.

%Funkce trajektorie z crawling ke zpátečce:
xC=crawling(end,1);yC=crawling(end,2);
xR=zpatecka(end,1);yR=zpatecka(end,2);
a=yR-yC;b=xC-xR;c=xR*(yC-yR)-yR*(xC-xR);
xRC=volba_N12_NRC(end,1);yRC=volba_N12_NRC(end,2);

```

```
d=(a*xRC+b*yRC+c)/sqrt(a^2+b^2);
sklon=xR-xC;
G_RC=(ws*sklon)^2+(wd*d)^2;

%Funkce trajektorie z prvního do druhého rychlostního stupně:
x2=druhy_rychl_stupen(end,1);y2=druhy_rychl_stupen(end,2);
x1=prvni_rychl_stupen(end,1);y1=prvni_rychl_stupen(end,2);
a=y2-y1;b=x1-x2;c=x2*(y1-y2)-y2*(x1-x2);
x12=druhy_rychl_stupen(1,1);y12=druhy_rychl_stupen(1,2);
d=(a*x12+b*y12+c)/sqrt(a^2+b^2);
sklon=x2-x1;
G_12=(ws*sklon)^2+(wd*d)^2;

%Funkce trajektorie z třetího do čtvrtého rychlostního stupně:
x4=ctvrty_rychl_stupen(end,1);y4=ctvrty_rychl_stupen(end,2);
x3=treti_rychl_stupen(end,1);y3=treti_rychl_stupen(end,2);
a=y4-y3;b=x3-x4;c=x4*(y3-y4)-y4*(x3-x4);
x34=ctvrty_rychl_stupen(1,1);y34=ctvrty_rychl_stupen(1,2);
d=(a*x34+b*y34+c)/sqrt(a^2+b^2);
sklon=x4-x3;
G_34=(ws*sklon)^2+(wd*d)^2;

%Funkce trajektorie z šestého do pátého rychlostního stupně:
x6=sesty_rychl_stupen(end,1);y6=sesty_rychl_stupen(end,2);
x5=paty_rychl_stupen(end,1);y5=paty_rychl_stupen(end,2);
a=y6-y5;b=x5-x6;c=x6*(y5-y6)-y6*(x5-x6);
x56=paty_rychl_stupen(1,1);y56=paty_rychl_stupen(1,2);
d=(a*x56+b*y56+c)/sqrt(a^2+b^2);
sklon=x6-x5;
G_56=(ws*sklon)^2+(wd*d)^2;

Ld=yC-yR;%Délka drah řazení.

%Cílová funkce:
F=(G_RC+G_12+G_34+G_56)/Ld^2;
```

## 11.11.2 Procedura definující rozměry pravostranného mechanismu

```
function rozmery
%Definuje rozměry pravostranného mechanismu.

%Délky jednotlivých členů mechanismu:
global L3k L4 gama psi L7 L8 L9 L11 L13
L3k=16;psi=6;L4=922;gama=7;L7=726;L8=50;L9=265;L11=265;L13=758;
gama=pi/180*(gama);psi=pi/180*(psi);

%Souřadnice bodů:
global DV4 D5 H5 O1 HV6 S8 S1 PV11 LV1 LS10 V1 PS10 R12 RS1 RK6 M12 %uprava2
DV4=[-64 842.5 -1.5];
D5=[61 0.0 0.0];
O1=[-34. 936. 41.5];
H5=[61 0.0 711.5];
HV6=[67.5 13.5 1.5];
RK6=[ 2 1089. 0.0];
S8=[-0.5 44.5 59];
S1=[-81.5 2069 768];
%upravalzacatek
LS10=[0.0 0.0 -100];
V1=[-81 2339.5 809];
PS10=[789 0.0 -99.0];
%upraval konec
```

```

R12=[0.0 5.5 -59.0];
RS1=[708 2069 768.5];
PV11=[-17.0 173.5 97.5];
LV1=[-67.5 2236.5 808];%uprava3
M12=[17.0 35.0 348.0];

%Definuje počáteční odhad vektoru q, který obsahuje hodnoty natočení a posuvu
%ve vazbách:
global q_poc_poloha
%Odhad vektoru q v počáteční poloze, tzn. p=0, fii=0:
q_poc_poloha=[ 0 %q(1)
               0 %q(2)
               0 %q(3)
               0 %q(4)
               0 %q(5)
               0 %q(6)
               0 %q(7)
               0 %q(8)
               0 %q(9)
               0 %q(10)
               0 %q(11)
               0 %q(12)
               0 %q(13)
               0 %q(14)
               0 %q(15)
               0 %q(16)
               0 %q(17)
               0 %q(18)
               0 %q(19)
               0 %q(20)
               0 %q(21)
               0 %q(22)
               0 %q(23)
               0 %q(24)
               0 %q(25)
               0 %q(26)
               0 %q(27)
               0 %q(28)
               0 %q(29)
               0 %q(30)
               0 %q(31)
               0 %q(32)
               0 %q(33)
               0 %q(34)
               0 %q(35)
               0]; %q(36)

```

### 11.11.3 Procedura sestavující vektor hodnot reziduí algebraických rovnic získaných ze základních smyček

```

function Rez=nonlin_system(q)
%Sestaví vektor Rez hodnot reziduí algebraických rovnic získaných ze základních
%smyček; aby byla splněna úlohy polohy, měl by být vektor nulový tudíž všechna
%rezidua rovny nule.

Rez=[ zakladni_smycka_I(q)
      zakladni_smycka_II(q)
      zakladni_smycka_III(q)
      zakladni_smycka_IV(q)
      zakladni_smycka_V(q)

```

---

`zakladni_smycka_VI(q)];`

### 11.11.4 Procedura základní smyčky I

```
function vektor_rc=zakladni_smycka_I(q)
%Sestaví maticovou rovnici základní smyčky I; vybere rezidua šesti nezávislých
%rovníc, které by měli být rovny nule, a ty seřadí do výsledného vektoru
%vektor_rc.

global p fii L3k L4 D5 gama O1 psi

T14=Ty(p)*Ry(fii)*Ry(-psi)*Rx(q(1))*Ty(L3k)*Rz(q(2));
T45=Ty(L4)*Rx(q(3))*Ry(q(4))*Rz(q(5))*T(-D5);
T51=Rz(q(6))*Ry(gama)*T(-O1);

A=T14*T45*T51-eye(4);

vektor_rc=[A(1,2);A(1,3);A(1,4);A(2,3);A(2,4);A(3,4)];
```

### 11.11.5 Procedura základní smyčky II

```
function vektor_rc=zakladni_smycka_II(q)
%Sestaví maticovou rovnici základní smyčky II; vybere rezidua šesti nezávislých
%rovníc, které by měli být rovny nule, a ty seřadí do výsledného vektoru
%vektor_rc.
global L4 D5 H5 HV6 L7 DV4

T45=Ty(L4)*Rx(q(3))*Ry(q(4))*Rz(q(5))*T(-D5);
T56=T(H5)*Rx(q(7))*Ry(q(8))*Rz(q(9));
T67=T(HV6)*Rx(q(10))*Ry(q(11))*Rz(q(12));
T74=Tz(-L7)*Rx(q(13))*Ry(q(14))*T(-DV4);

A=T45*T56*T67*T74-eye(4);
vektor_rc=[A(1,2);A(1,3);A(1,4);A(2,3);A(2,4);A(3,4)];
```

### 11.11.6 Procedura základní smyčky III

```
function vektor_rc=zakladni_smycka_III(q)
%Sestaví maticovou rovnici základní smyčky III; vybere rezidua šesti nezávislých
%rovníc, které by měli být rovny nule, a ty seřadí do výsledného vektoru
%vektor_rc.

global O1 gama H5 RK6 S8 S1

T15=T(O1)*Ry(-gama)*Rz(-q(6));
T56=T(H5)*Rx(q(7))*Ry(q(8))*Rz(q(9));
T68=T(RK6)*Rx(q(15));
T81=T(S8)*Rx(q(16))*Ry(q(17))*Rz(q(18))*T(-S1);

A=T15*T56*T68*T81-eye(4);

vektor_rc=[A(1,2);A(1,3);A(1,4);A(2,3);A(2,4);A(3,4)];
```

### 11.11.7 Procedura základní smyčky IV

```
function vektor_rc=zakladni_smycka_IV(q)
%Sestaví maticovou rovnici základní smyčky IV; vybere rezidua šesti nezávislých
%rovníc, které by měli být rovny nule, a ty seřadí do výsledného vektoru
%vektor_rc.
```

```
global S1 S8 L8 L9 LS10 V1

T18=T(S1)*Rz(-q(18))*Ry(-q(17))*Rx(-q(16))*T(-S8);
T89=Ty(L8)*Rx(q(19))*Ry(q(20))*Rz(q(21));
T9_10=Ty(L9)*Rx(q(22))*T(-LS10);
T10_1=Tx(q(23))*Rx(q(24))*T(-V1);

A=T18*T89*T9_10*T10_1-eye(4);

vektor_rc=[A(1,2);A(1,3);A(1,4);A(2,3);A(2,4);A(3,4)];
```

### 11.11.8 Procedura základní smyčky V

```
function vektor_rc=zakladni_smycka_V(q)
%Sestaví maticovou rovnici základní smyčky V; vybere rezidua šesti nezávislých
%rovníc, které by měli být rovny nule, a ty seřadí do výsledného vektoru
%vektor_rc.
```

```
global V1 PS10 L11 R12 RS1

T1_10=T(V1)*Rx(-q(24))*Tx(-q(23));
T10_11=T(PS10)*Rx(q(25))*Ry(q(26))*Rz(q(27))*Ty(-L11);
T11_12=Rx(q(28))*T(-R12);
T12_1=Rx(q(29))*Ry(q(30))*Rz(q(31))*T(-RS1);

A=T1_10*T10_11*T11_12*T12_1-eye(4);

vektor_rc=[A(1,2);A(1,3);A(1,4);A(2,3);A(2,4);A(3,4)];
```

### 11.11.9 Procedura základní smyčky VI

```
function vektor_rc=zakladni_smycka_VI(q)
%Sestaví maticovou rovnici základní smyčky VI; vybere rezidua šesti nezávislých
%rovníc, které by měli být rovny nule, a ty seřadí do výsledného vektoru
%vektor_rc.
```

```
global PV11 L13 LV1 RS1 R12

T1_12=T(RS1)*Rz(-q(31))*Ry(-q(30))*Rx(-q(29));
T12_11=T(R12)*Rx(-q(28));
T11_13=T(PV11)*Rx(q(32))*Ry(q(33))*Rz(q(34))*Tx(-L13);
T13_1=Ry(q(35))*Rz(q(36))*T(-LV1);

A=T1_12*T12_11*T11_13*T13_1-eye(4);

vektor_rc=[A(1,2);A(1,3);A(1,4);A(2,3);A(2,4);A(3,4)];
```



## 11.12 Úloha polohy variantního návrhu A pravostranného mechanismu řazení

### 11.12.1 Hlavní procedura

Jako hlavní proceduru lze použít algoritmus uvedený v příloze 11.10.1.

### 11.12.2 Procedura definující rozměry pravostranného mechanismu variantního návrhu A

Rozměry variantního návrhu lze definovat též pomocí již uvedeného algoritmu v příloze 11.11.2. Pouze je třeba změnit část procedury začínající značkou `%upravalzacatek` a končící `%upravalkonec` následovně:

```
%upravalzacatek
LS10=[-394.5 0 -100-200];
SYM1=[313.5 2339.5 809+200];
PS10=[394.5 0.0 -99.0-200];
%upravalkonec
```

A dále je třeba doplnit v místě označeném `%uprava2` definici globální proměnné pomocí SYM1.

### 11.12.3 Procedura sestavující vektor hodnot reziduí algebraických rovnic získaných ze základních smyček

Lze použít beze změny proceduru z přílohy 11.11.3.

### 11.12.4 Procedura základní smyčky I

Pro sestavení maticové rovnice základní smyčky I a pro následovný výběr reziduí šesti nezávislých algebraických rovnic lze použít procedury uvedené v příloze 11.11.4.

#### 1.1.1 Procedura základní smyčky II

Pro sestavení maticové rovnice základní smyčky II a pro následovný výběr reziduí šesti nezávislých algebraických rovnic lze použít procedury uvedené v příloze 11.11.5.

#### 1.1.2 Procedura základní smyčky III

Pro sestavení maticové rovnice základní smyčky III a pro následovný výběr reziduí šesti

nezávislých algebraických rovnic lze použít procedury uvedené v příloze 11.11.6.

### 1.1.3 Procedura základní smyčky IV

```
function vektor_rc=zakladni_smycka_IV(q)
%Sestaví maticovou rovnici základní smyčky IV; vybere rezidua šesti nezávislých
%rovníc, které by měli být rovny nule, a ty seřadí do výsledného vektoru
%vektor_rc.

global S1 S8 L8 L9 LS10 SYM1

T18=T(S1)*Rz(-q(18))*Ry(-q(17))*Rx(-q(16))*T(-S8);
T89=Ty(L8)*Rx(q(19))*Ry(q(20))*Rz(q(21));
T9_10=Ty(L9)*Rx(q(22))*T(-LS10);
T10_1=Ty(q(23))*Ry(q(24))*T(-SYM1);

A=T18*T89*T9_10*T10_1-eye(4);

vektor_rc=[A(1,2);A(1,3);A(1,4);A(2,3);A(2,4);A(3,4)];
```

### 1.1.4 Procedura základní smyčky V

```
function vektor_rc=zakladni_smycka_V(q)
%Sestaví maticovou rovnici základní smyčky V; vybere rezidua šesti nezávislých
%rovníc, které by měli být rovny nule, a ty seřadí do výsledného vektoru
%vektor_rc.

global SYM1 PS10 L11 R12 RS1

T1_10=T(SYM1)*Ry(-q(24))*Ty(-q(23));
T10_11=T(PS10)*Rx(q(25))*Ry(q(26))*Rz(q(27))*Ty(-L11);
T11_12=Rx(q(28))*T(-R12);
T12_1=Rx(q(29))*Ry(q(30))*Rz(q(31))*T(-RS1);

A=T1_10*T10_11*T11_12*T12_1-eye(4);

vektor_rc=[A(1,2);A(1,3);A(1,4);A(2,3);A(2,4);A(3,4)];
```

### 1.1.5 Procedura základní smyčky VI

Pro sestavení maticové rovnice základní smyčky VI a pro následovný výběr reziduí šesti nezávislých algebraických rovnic lze použít procedury uvedené v příloze 11.11.9.

## 11.13 Úloha polohy variantního návrhu B pravostranného mechanismu řazení

### 11.13.1 Hlavní procedura variantního návrhu B

Jako hlavní proceduru lze použít algoritmus uvedený v příloze 11.11.1.

### 11.13.2 Procedura definující rozměry pravostranného mechanismu variantního návrhu B

Rozměry variantního návrhu lze definovat též pomocí již uvedeného algoritmu v příloze 11.11.2. Pouze je třeba změnit část procedury začínající značkou `%upravalzacatek` a končící `%upravalkonec` následovně:

```
%upravalzacatek
LS10=[-394.5 0 -100];
SYM1=[313.5 2339.5 809];
PS10=[394.5 0.0 -99.0];
%upravalkonec
```

Řádek označený `%uprava3` je zapotřebí přepsat následovně: `LV9=[17.0 173.5 97.5];`

A dále je třeba doplnit v místě označeném `%uprava2` definice globální proměnné pomocí `SYM1` a `LV9`.

### 11.13.3 Procedura sestavující vektor hodnot reziduí algebraických rovnic získaných ze základních smyček

Lze použít beze změny proceduru z přílohy 11.11.3.

### 11.13.4 Procedura základní smyčky I

Pro sestavení maticové rovnice základní smyčky I a pro následovný výběr reziduí šesti nezávislých algebraických rovnic lze použít procedury uvedené v příloze 11.11.4.

### 11.13.5 Procedura základní smyčky II

Pro sestavení maticové rovnice základní smyčky II a pro následovný výběr reziduí šesti nezávislých algebraických rovnic lze použít procedury uvedené v příloze 11.11.5.

### 11.13.6 Procedura základní smyčky III

Pro sestavení maticové rovnice základní smyčky III a pro následovný výběr reziduí šesti nezávislých algebraických rovnic lze použít procedury uvedené v příloze 11.11.6.

### 11.13.7 Procedura základní smyčky IV

```
function vektor_rc=zakladni_smycka_IV(q)
%Sestaví maticovou rovnici základní smyčky IV; vybere rezidua šesti nezávislých
%rovnic, které by měli být rovny nule, a ty seřadí do výsledného vektoru
%vektor_rc.
```

```
global S1 S8 L8 L9 LS10 SYM1
```

```
T18=T(S1)*Rz(-q(18))*Ry(-q(17))*Rx(-q(16))*T(-S8);
T89=Ty(L8)*Rx(q(19))*Rz(q(21));
T9_10=Ty(L9)*Rx(q(22))*Ry(q(20))*Rz(q(23))*T(-LS10);
T10_1=Rz(q(24))*T(-SYM1);
```

```
A=T18*T89*T9_10*T10_1-eye(4);
```

```
vektor_rc=[A(1,2);A(1,3);A(1,4);A(2,3);A(2,4);A(3,4)];
```

### 11.13.8 Procedura základní smyčky V

```
function vektor_rc=zakladni_smycka_V(q)
%Sestaví maticovou rovnici základní smyčky V; vybere rezidua šesti nezávislých
%rovnic, které by měli být rovny nule, a ty seřadí do výsledného vektoru
%vektor_rc.
```

```
global SYM1 PS10 L11 R12 RS1
```

```
T1_10=T(SYM1)*Rz(-q(24));
T10_11=T(PS10)*Rx(q(25))*Ry(q(26))*Rz(q(27))*Ty(-L11);
T11_12=Ry(pi/180*(-0))*Rx(q(28))*Ry(-pi/180*(0))*T(-R12);
T12_1=Rx(q(29))*Ry(q(30))*Rz(q(31))*T(-RS1);
```

```
A=T1_10*T10_11*T11_12*T12_1-eye(4);
```

```
vektor_rc=[A(1,2);A(1,3);A(1,4);A(2,3);A(2,4);A(3,4)];
```

### 11.13.9 Procedura základní smyčky VI

```
function vektor_rc=zakladni_smycka_VI(q)
%Sestaví maticovou rovnici základní smyčky VI; vybere rezidua šesti nezávislých
%rovnic, které by měli být rovny nule, a ty seřadí do výsledného vektoru
%vektor_rc.
```

```
global PV11 L13 L9 LS10 PS10 L11 LV9
```

```
T9_10=Ty(L9)*Rx(q(22))*Ry(q(20))*Rz(q(23))*T(-LS10);
T10_11=T(PS10)*Rx(q(25))*Ry(q(26))*Rz(q(27))*Ty(-L11);
T11_13=T(PV11)*Rx(q(32))*Ry(q(33))*Rz(q(34))*Tx(-L13);
T13_9=Ry(q(35))*Rz(q(36))*T(-LV9);
```

```
A=T9_10*T10_11*T11_13*T13_9-eye(4);
```

```
vektor_rc=[A(1,2);A(1,3);A(1,4);A(2,3);A(2,4);A(3,4)];
```

## 11.14 Úloha polohy variantního návrhu C pravostranného mechanismu řazení

### 11.14.1 Hlavní procedura variantního návrhu C

Jako hlavní proceduru lze použít algoritmus uvedený v příloze 11.11.1.

### 11.14.2 Procedura definující rozměry pravostranného mechanismu variantního návrhu C

```
function rozmery
%Definuje rozměry pravostranného mechanismu variantní návrh C.

%Délky jednotlivých členů mechanismu:
global L3k L4 gama psi L7 L11 L13 L6
L3k=16;psi=6;L4=922;gama=7;L7=726;L11=265;L13=754.2;
gama=pi/180*(gama);psi=pi/180*(psi);L6=1089+50+265;

%Souřadnice bodů:
global DV4 D5 H5 O1 HV6 PV11 LV6 LS10 PS10 R12 RS1 M12 SYM1
DV4=[-64 842.50 -1.5];
D5=[ 61 0.0 0.0];
O1=[ -34. 936. 41.5];
H5=[61 0.0 711.5];
HV6=[67.5 13.5 1.5];
LS10=[-394.5 0 -100-000];
SYM1=[313.5 2339.5 809+00];
PS10=[ 394.5 0.0 -99.0-00];
R12=[0.0 5.5 59.0];
RS1=[ 708 2069 768.5-2*59];
PV11=[ -17.0 173.5 97.5-0];
LV6=[ 17.0 1089+50+173.5 97.5-0];
M12=[17.0 35.0 348.0];

%Definuje počáteční odhad vektoru q, který obsahuje hodnoty natočení ve
%vazbách:
global q_poc_poloha
%Odhad vektoru q v počáteční poloze, tzn. p=0, fii=0:
q_poc_poloha=[ 0 %q(1)
               0 %q(2)
               0 %q(3)
               0 %q(4)
               0 %q(5)
               0 %q(6)
               0 %q(7)
               0 %q(8)
               0 %q(9)
               0 %q(10)
               0 %q(11)
               0 %q(12)
               0 %q(13)
               0 %q(14)
               0 %q(15)
               0 %q(16)]
```

```

0 %q(17)
0 %q(18)
0 %q(19)
0 %q(20)
0 %q(21)
0 %q(22)
0 %q(23)
0 %q(24)
0 %q(25)
0 %q(26)
0 %q(27)
0 %q(28)
0 %q(29)
0]; %q(30)

```

### 11.14.3 Procedura sestavující vektor hodnot reziduí algebraických rovnic získaných ze základních smyček

```

function Rez=soustava_transc_rc(q)
%Sestaví vektor Rez hodnot reziduí algebraických rovnic získaných ze základních
%smyček; aby byla splněna úlohy polohy, měl by být vektor nulový tudíž všechna
%rezidua rovny nule.

Rez=[ zakladni_smycka_I(q)
      zakladni_smycka_II(q)
      zakladni_smycka_III(q)
      zakladni_smycka_IV(q)
      zakladni_smycka_V(q)];

```

### 11.14.4 Procedura základní smyčky I

Pro sestavení maticové rovnice základní smyčky I a pro následovný výběr reziduí šesti nezávislých algebraických rovnic lze použít procedury uvedené v příloze 11.11.4.

### 11.14.5 Procedura základní smyčky II

Pro sestavení maticové rovnice základní smyčky II a pro následovný výběr reziduí šesti nezávislých algebraických rovnic lze použít procedury uvedené v příloze 11.11.5.

### 11.14.6 Procedura základní smyčky III

```

function vektor_rc=zakladni_smycka_III(q)
%Sestaví maticovou rovnici základní smyčky III; vybere rezidua šesti nezávislých
%rovnic, které by měli být rovny nule, a ty seřadí do výsledného vektoru
%vektor_rc.
global L6 LS10 SYM1 O1 gama H5

T15=T(O1)*Ry(-gama)*Rz(-q(6));
T56=T(H5)*Rx(q(7))*Ry(q(8))*Rz(q(9));
T6_10=Ty(L6)*Rx(q(15))*Ry(q(16))*Rz(q(17))*T(-LS10);
T10_1=Rx(-pi/180*(2))*Rz(q(18))*Rx(pi/180*(2))*T(-SYM1);

A=T15*T56*T6_10*T10_1-eye(4);
vektor_rc=[A(1,2);A(1,3);A(1,4);A(2,3);A(2,4);A(3,4)];

```

### 11.14.7 Procedura základní smyčky IV

```
function vektor_rc=zakladni_smycka_IV(q)
%Sestaví maticovou rovnici základní smyčky IV; vybere rezidua šesti nezávislých
%rovníc, které by měli být rovny nule, a ty seřadí do výsledného vektoru
%vektor_rc.

global SYM1 PS10 L11 R12 RS1

T1_10=T(SYM1)*Rx(-pi/180*(0))*Rz(-q(18))*Rx(pi/180*(0));
T10_11=T(PS10)*Rx(q(19))*Ry(q(20))*Rz(q(21))*Ty(-L11);
T11_12=Ry(-pi/180*(0))*Rx(q(22))*Ry(pi/180*(0))*T(-R12);
T12_1=Rx(q(23))*Ry(q(24))*Rz(q(25))*T(-RS1);

A=T1_10*T10_11*T11_12*T12_1-eye(4);

vektor_rc=[A(1,2);A(1,3);A(1,4);A(2,3);A(2,4);A(3,4)];
```

### 11.14.8 Procedura základní smyčky V

```
function vektor_rc=zakladni_smycka_V(q)
%Sestaví maticovou rovnici základní smyčky V; vybere rezidua šesti nezávislých
%rovníc, které by měli být rovny nule, a ty seřadí do výsledného vektoru
%vektor_rc.

global PV11 L13 LS10 PS10 L11 LV6 L6

T6_10=Ty(L6)*Rx(q(15))*Ry(q(16))*Rz(q(17))*T(-LS10);
T10_11=T(PS10)*Rx(q(19))*Ry(q(20))*Rz(q(21))*Ty(-L11);
T11_13=T(PV11)*Rx(q(26))*Ry(q(27))*Rz(q(28))*Tx(-L13);
T13_6=Ry(q(29))*Rz(q(30))*T(-LV6);

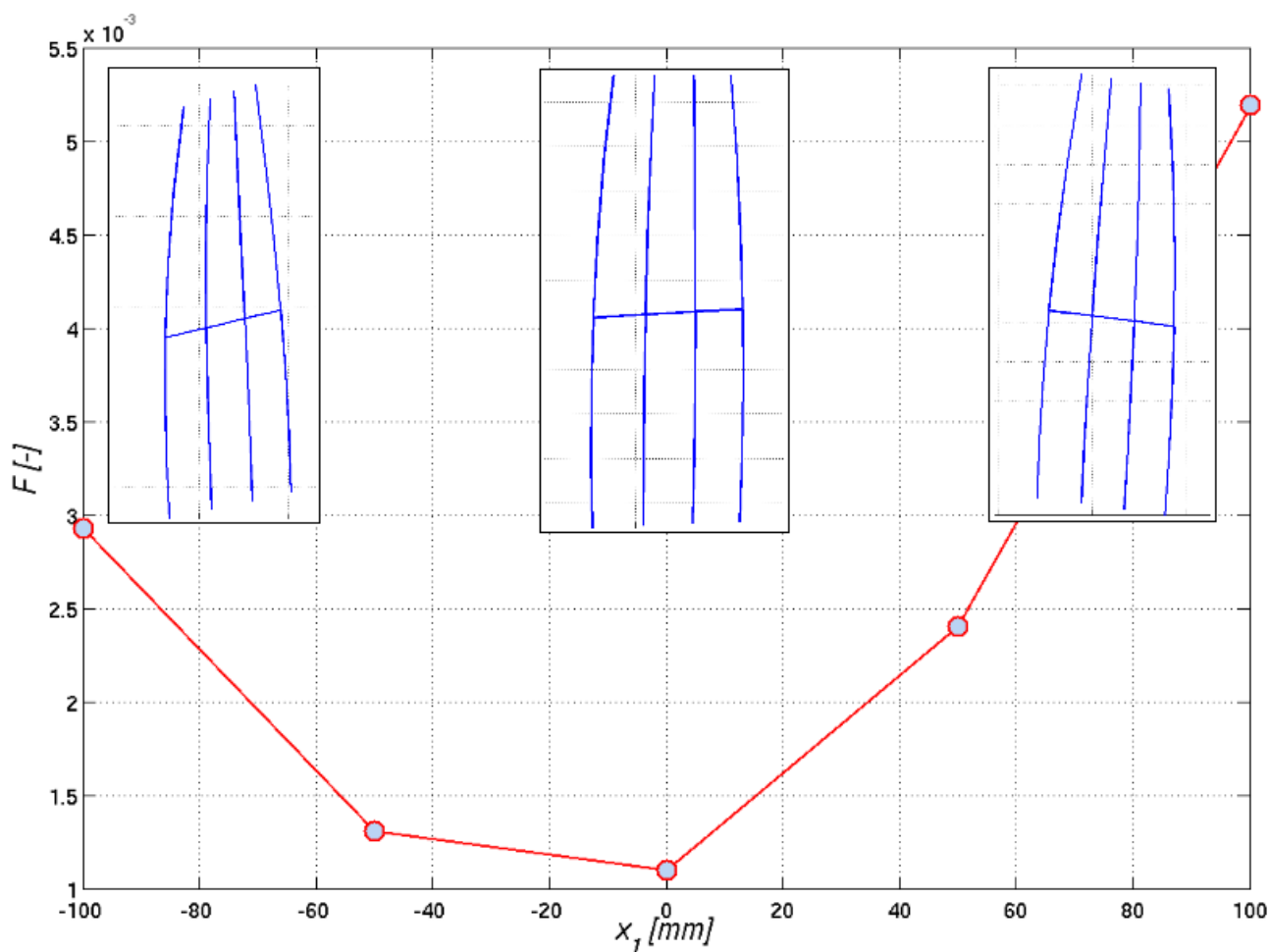
A=T6_10*T10_11*T11_13*T13_6-eye(4);

vektor_rc=[A(1,2);A(1,3);A(1,4);A(2,3);A(2,4);A(3,4)];+
```

## 11.15 Citlivostní analýzy pravostranného mechanismu řazení

### 11.15.1 Citlivost na posuv kabiny nákladního vozu

Kabina nákladního vozu je ukotvena k rámu ve čtyřech bodech, v přední části kabiny pomocí dvou čepů a v zadní části je usazena v pryžových pouzdrech. Ze způsobu ukotvení vyplývá, že k posuvu kabiny vůči rámu vozidla dochází i při běžném provozu, další rozměrové odchylky vznikají při ustavení kabiny během montáže. Proto je logické začít analýzou citlivosti mechanismu řazení na změnu polohy kabiny vůči rámu vozidla. Na kabině vozidla jsou umístěny body mechanismu  $S_I$ ,  $V_I$ ,  $LV_I$  a  $RS_I$ , viz. kap. 8.4. Citlivostní analýza bude spočívat v současném posuvu všech zmíněných bodů současně ve směru jednotlivých os pevného souřadného systému 1.

Změna polohy kabiny ve směru  $x_1$ Obrázek 11.3: Citlivost na změnu polohy kabiny ve směru  $x_1$ 

Na výše uvedeném obr. 11.3 je zobrazena citlivost pravostranného mechanismu na posuv kabiny ve směru osy  $x_1$ , na vertikální ose je vykreslena hodnota cílové funkce a na horizontální ose je vynesena posuv kabiny.  $x_1 = 0$  odpovídá neposunuté kabině,  $x_1 = 100 \text{ mm}$  odpovídá posunutí všech bodů kabiny o  $100 \text{ mm}$  v kladném směru osy  $x_1$ . Dále jsou v grafu vykresleny dráhy řazení pro krajní body a původní nastavení parametru  $x_1 = -100; 0; 100 \text{ mm}$ . Vypočtené hodnoty jsou také shrnuty v tab. 11.2. Z grafu i tabulky je zřejmé, že mechanismus řazení je netečný na změnu polohy kabiny ve směru  $x_1$  v rámci  $\pm 100 \text{ mm}$ .

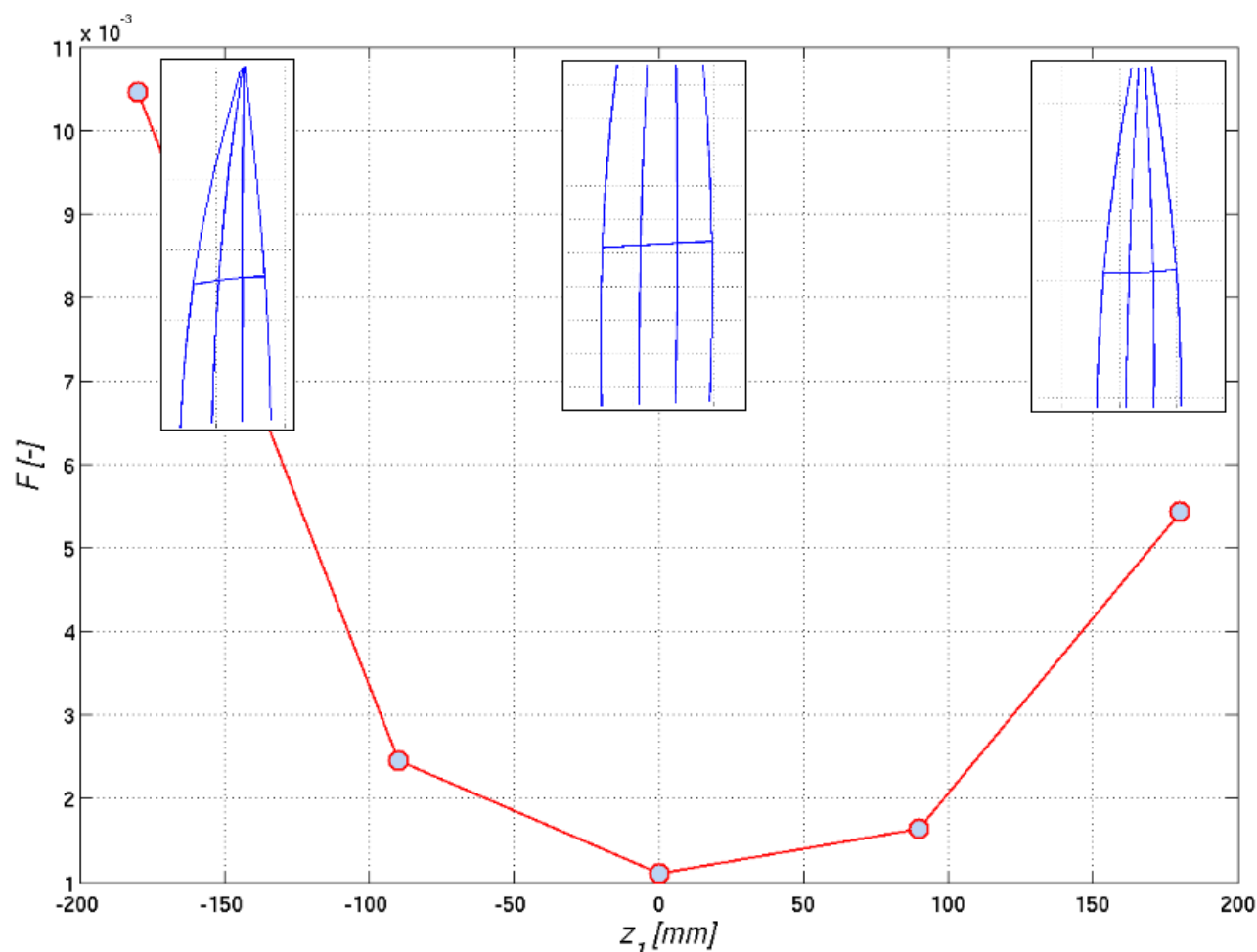
Tabulka 11.2: Hodnoty cílové funkce v závislosti na posuvu kabiny ve směru  $x_1$ 

$x_1 [\text{mm}]$	-100	-50	0	50	100
$F \cdot 10^4 [-]$	29	13	11	24	52



## Změna polohy kabiny ve směru $z_1$

Citlivost mechanismu na změnu polohy kabiny ve směru osy  $z_1$  je uvedena v následujícím obr. 11.4 a tab. 11.3. Dráhy řazení jsou odolné vůči změně tohoto parametru v rozsahu cca  $\pm 100 \text{ mm}$ .



Obrázek 11.4: Citlivost na změnu polohy kabiny ve směru  $z_1$

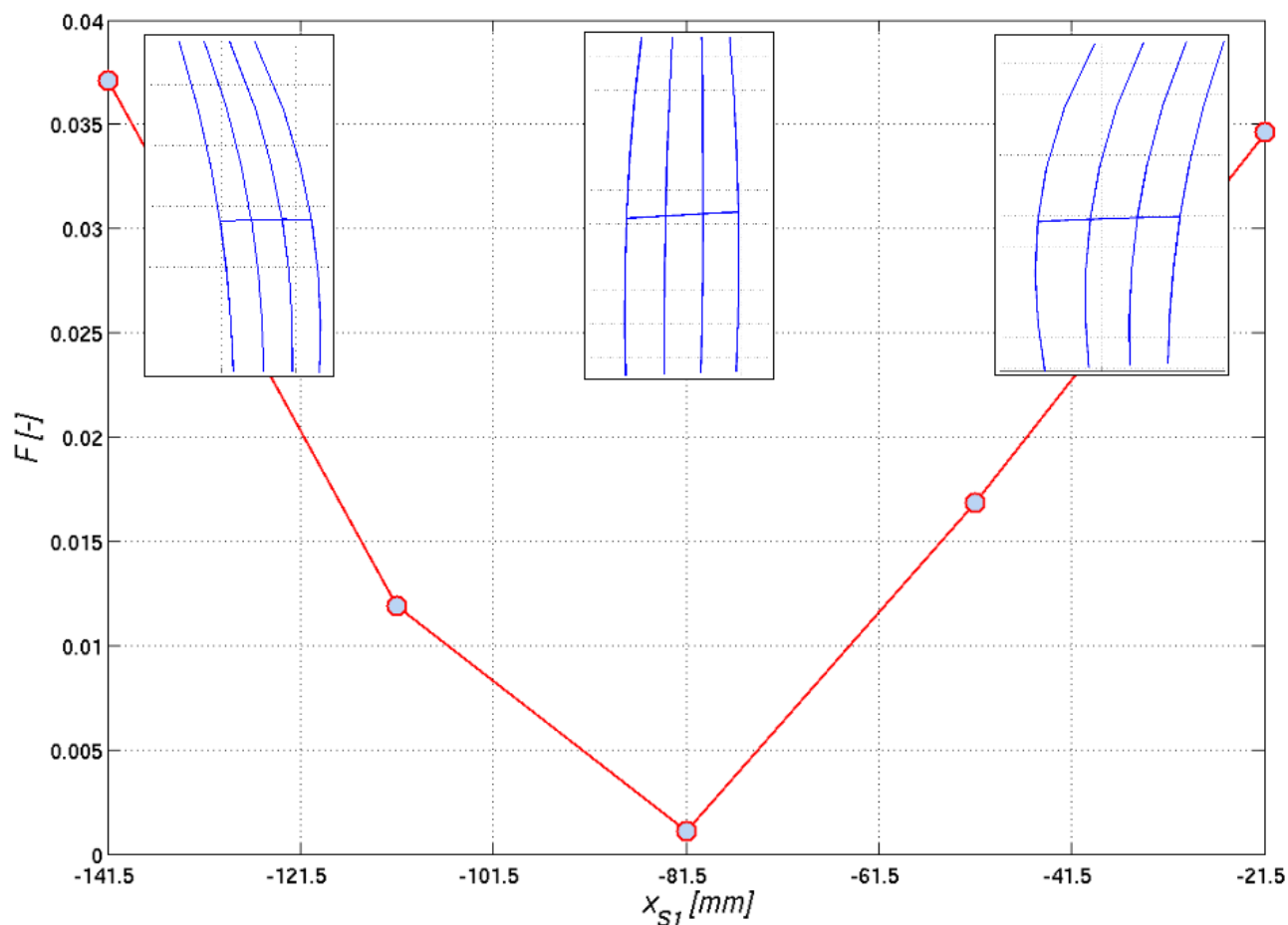
Tabulka 11.3: Hodnoty cílové funkce v závislosti na posuvu kabiny ve směru  $z_1$

$z_1$ [mm]	-180	-90	0	90	180
$F \cdot 10^4$ [-]	105	25	11	16	54

### 11.15.2 Změna polohy bodu $S_1$

#### Posuv bodu $S_1$ ve směru $x_1$

Analýza se zabývá posuvem bodu  $S_1$ , viz obr. 8.10, který leží ve středu sférické vazby, pomocí které je k rámu 1 vázán člen 8, ve směru  $x_1$ . Původní souřadnice má hodnotu  $x_{S1} = -81,5 \text{ mm}$ . Opět je přiložen obr. 11.5 a tab. 11.4.

Obrázek 11.5: Citlivost na změnu polohy bodu  $S_1$  ve směru  $x_1$ Tabulka 11.4: Hodnoty cílové funkce v závislosti na posuvu bodu  $S_1$  ve směru  $x_1$ 

$x_{S1}$ [mm]	-141,5	-111,5	-81,5	-51,5	-21,5
$F \cdot 10^4$ [-]	371	119	11	169	346

### Posuv bodu $S_1$ ve směru $y_1$

Citlivost mechanismu na posuv bodu  $S_1$  ve směru  $y_1$  není třeba analyzovat, protože délku členu 6 vyjádřenou pomocí souřadnice  $x_{RK6}$  (viz obr. 8.10) je možné dle potřeby doladit během montáže pomocí šroubového spojení viz obr. 8.1, čímž je eliminován vliv posuvu bodu  $S_1$  ve směru  $y_1$ .

### Posuv bodu $S_1$ ve směru $z_1$

Citlivostní analýza zkoumá vliv posuvu bodu  $S_1$  ve směru  $z_1$ , původní souřadnice má hodnotu  $z_{S1} = 2069$  mm. Změna této souřadnice v rámci  $\pm 80$  mm nemá na dráhy řazení téměř žádný dopad, proto není třeba dokládat graf ani tabulku.

### 11.15.3 Změna polohy bodu $V_1$

#### Posuv bodu $V_1$ ve směru $y_1$

Jedná se o analýzu vlivu posunutí osy válcové vazby, jenž váže trámec 10 k rámu 1, ve směru  $y_1$ . Původní rozměr je  $y_{V1} = 2339,5 \text{ mm}$ , bylo zjištěno, že pokud je souřadnice v rozmezí  $y_{V1} = \langle 2240; 2400 \rangle \text{ mm}$ , je vliv na tvar drah řazení malý.

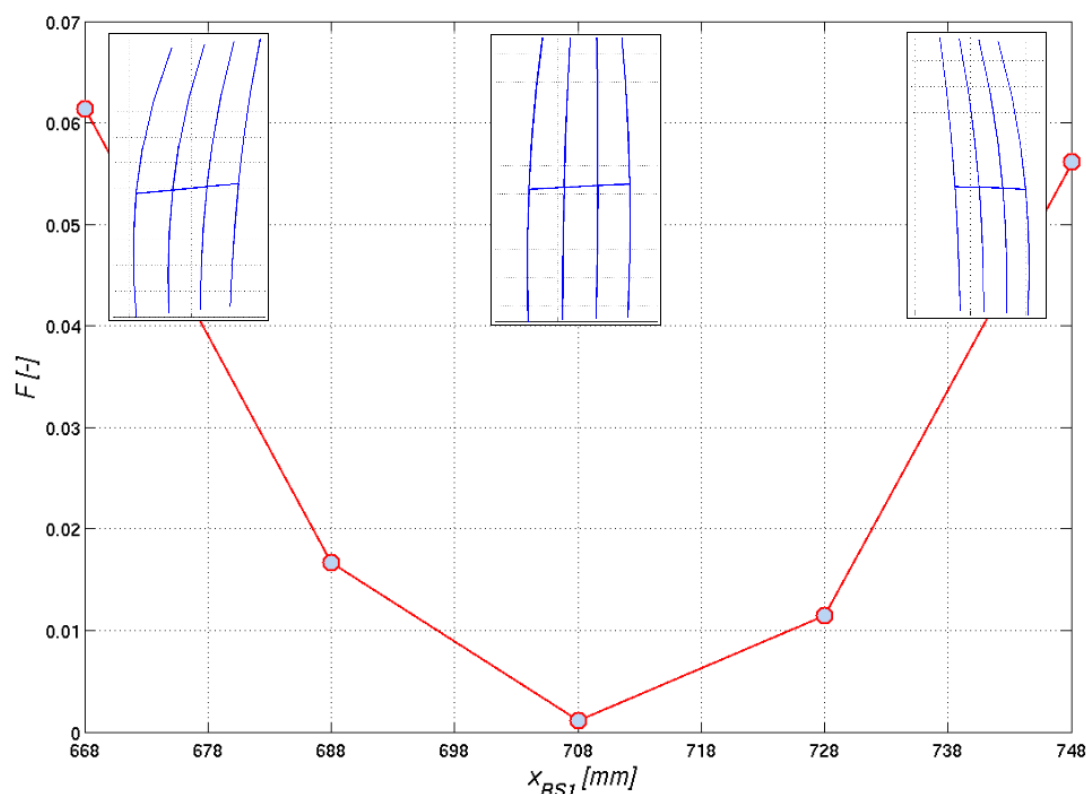
#### Posuv bodu $V_1$ ve směru $z_1$

Původní rozměr této souřadnice je  $z_{V1} = 809 \text{ mm}$ , změna této souřadnice v rámci cca  $\pm 100 \text{ mm}$  nemá na dráhy řazení téměř žádný vliv.

### 11.15.4 Změna polohy bodu $RS_1$

#### Posuv bodu $RS_1$ ve směru $x_1$

Posledním zkoumaným parametrem je poloha středu sférického kloubu  $RS_1$ , který váže řadicí páku 12 k rámu 1, původní souřadnice měla hodnotu  $x_{RS1} = 708 \text{ mm}$ . Z obr. 11.6 je zřejmé, že posuv bodu ve směru osy  $x_1$  způsobuje naklopení drah řazení.



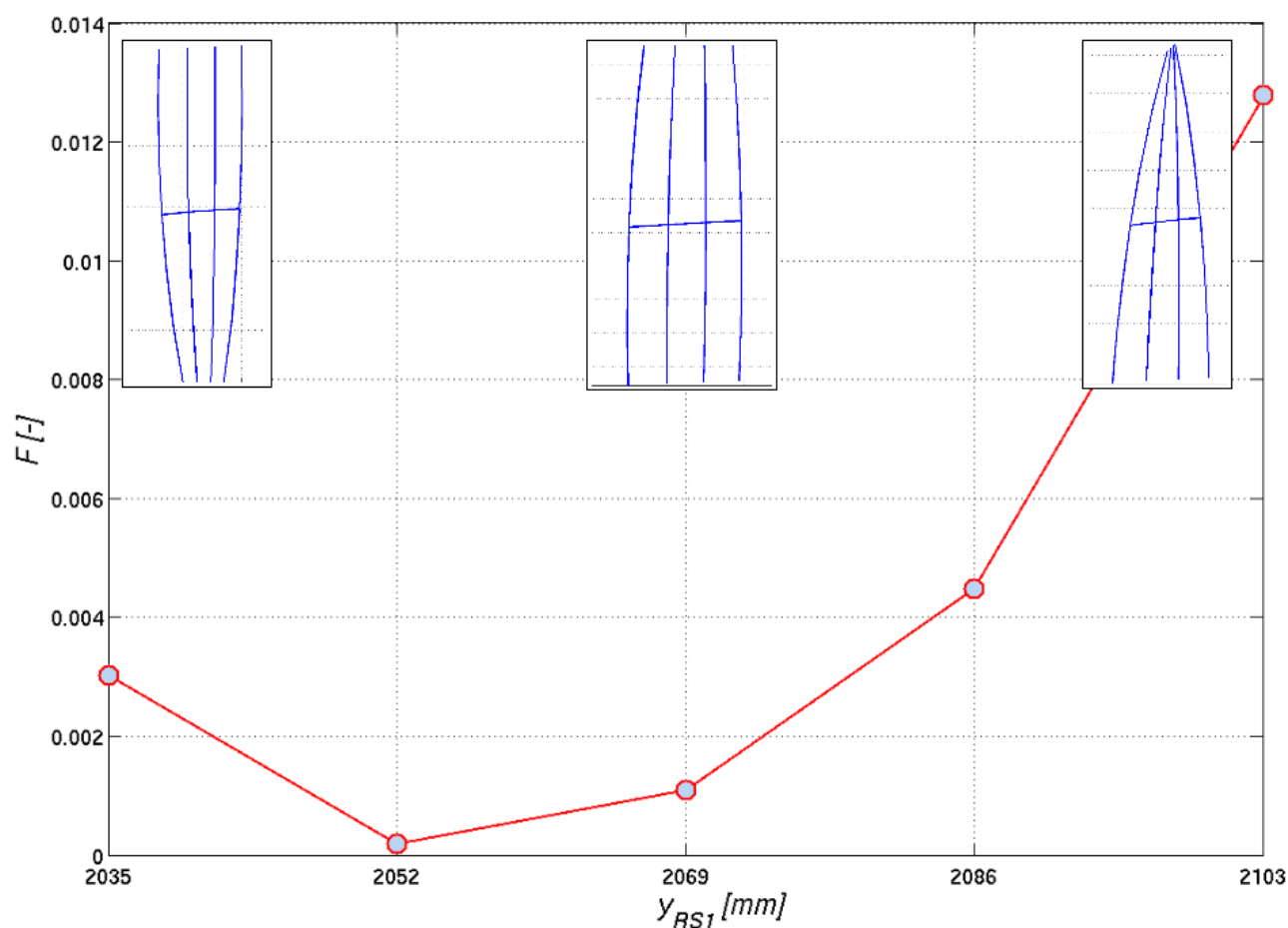
Obrázek 11.6: Citlivost na změnu polohy bodu  $RS_1$  ve směru  $x_1$

Tabulka 11.5: Hodnoty cílové funkce v závislosti na posuvu bodu  $RS_1$  ve směru  $x_1$ 

$x_{RS1}$ [mm]	668	688	708	728	748
$F \cdot 10^4$ [-]	613	167	11	114	562

**Posuv bodu  $RS_1$  ve směru  $y_1$** 

Původní souřadnice měla hodnotu  $y_{RS1} = 2069$  mm, obdobně jako v případě minulé analýzy dochází v tomto případě k deformování drah řazení podle toho, kterým směrem posouváme bod, viz obr. 11.7 a tab. 11.6. Dráhy se jakoby uzavírají buď v horní nebo dolní části.

Obrázek 11.7: Citlivost na změnu polohy bodu  $RS_1$  ve směru  $y_1$ Tabulka 11.6: Hodnoty cílové funkce v závislosti na posuvu bodu  $RS_1$  ve směru  $y_1$ 

$y_{RS1}$ [mm]	2035	2052	2069	2086	2103
$F \cdot 10^4$ [-]	30	2	11	45	128

**Posuv bodu  $RS_1$  ve směru  $z_1$** 

Změna tohoto parametru, jehož původní hodnota je  $z_{RS1} = 768,5$  mm, v rámci  $\pm 80$  mm nemá na tvar drah výraznější dopad.

## **11.16 Rešerše stavu poznání v problematice analýzy a syntézy mechanismů, matematické teorie optimalizace a řešení soustav nelineárních, algebraických rovnic**

Následující rešerše podává přehled o stavu poznání ve čtyřech podstatných oblastech, a to analýza mechanismů, syntéza mechanismů, matematická teorie optimalizace a řešení soustav nelineárních, algebraických rovnic. Odborné texty jsou rozřazeny dle těchto oblastí. Rešerše je vyhodnocena v kap. 2

### **11.16.1 Analýza a syntéza mechanismů**

[7] LEDERER, P. *Teorie a optimalizace mechanických systémů I*. Praha : České vysoké učení technické v Praze, 1988. 143 s.

[8] LEDERER, P. *Teorie a optimalizace mechanických systémů II*. Praha : České vysoké učení technické v Praze, 1989. 70 s.

Autor se zabývá použitím optimalizačních metod při syntéze mechanismů. V první části knihy jsou rozebrány různé optimalizační metody z matematického pohledu. Jsou popsány metody jednorozměrného vyhledávání minima, optimalizace bez a s omezením. Bohužel kniha je z roku 1989 a optimalizační metody zaznamenaly od této doby silný pokrok. Ku příkladu zcela chybí metoda Broyden–Fletcher–Goldfarb–Shanno, což jedna z nejpoužívanějších metod optimalizace bez omezení. V kapitole o optimalizaci s omezením chybí taktéž často používaná metoda rozšířených Lagrangianů. V druhé části knihy jsou uvedeny příklady syntézy několika rovinných mechanismů. Bohužel pouze u prvního příkladu jsou uvedeny numerické výsledky optimalizačního procesu. U ostatních příkladů je vždy uvedeno pouze kinematické schéma, kinematické a popřípadě pohybové rovnice, formulace cílové funkce a vedlejších podmínek. Zcela zde chybí aplikace nebo srovnání popsaných metod optimalizace. Kniha [2] je pokračováním publikace [1]. Zabývá se syntézou mechanismů metodami, které nevyužívají matematické metody optimalizace. První část knihy uvádí grafické metody, které se dnes již nevyužívají. Dále se zabývá syntézou mechanismu pomocí tzv. „bodů přesnosti“ kolokační metodou. Tato metoda je vzhledem k použití optimalizačních metod ve spojení s moderní výpočetní technikou taktéž zastaralá. Použití komplexních čísel, které je taktéž rozebráno, je bohužel použitelné pouze pro rovinné mechanismy. V poslední části knihy jsou uvedeny postupy pro dimenzování vaček.

[6] BRÁT, V. *Maticové metody v analýze a syntéze prostorových vázaných systémů*.

Praha : Academia, 1981. 212 s.

Dílo do detailů popisuje kinematickou analýzu prostorových mechanismů pomocí maticové metody. Popis statické a dynamické silové analýzy je zkráceně zpracován taktéž. Kniha je určena spíše pro akademické pracovníky. Pro běžnou praxi chybí stručný a srozumitelný přehled kinematické nebo silové analýzy mechanismů. Dále kniha uvádí dnes již zastaralou metodu výpočtu úlohy polohy. Taktéž metody syntézy jsou již zastaralé. Kniha postrádá větší počet příkladů použití popisovaných metod. Například chybí příklad analýzy složeného mechanismu.

[9] ŽMINDÁK, M, et al. *Optimalizácia mechanických sústav*. Žilina : EDIS ŽU, 200. 261 s.

Kniha shrnuje v úvodní části popis optimalizačních metod a algoritmů. Cenné jsou numerické testy, které jsou uvedeny na konci každého pojednání o dané metodě. Čtenář si tedy může udělat představu o poměru efektivity a pracnosti sestavení algoritmu metody. Další části knihy se zabývají praktickými aplikacemi optimalizačních algoritmů. Uvedeno je taktéž pár jednoduchých příkladů optimalizace rovinných mechanismů s tuhými i elastickými členy. K tomuto je přiložen také zdrojový kód výpočtu v softwaru MATLAB.

[10] SÁGA, M., et al. *Aplikácia optimalizačných algoritmov v mechanike telies*. Žilina : VTS pri ŽU, 2006. 240 s. ISBN 80-969165-9-9.

V úvodu knihy jsou shrnuty metody optimalizace a bez popisu algoritmů těchto metod. Zbylá část díla se zabývá praktickými příklady, které jsou z oblasti mechaniky poddajných těles. Praktické příklady optimalizace mechanismů v knize nejsou obsaženy. Při uvažování mechanismů s elastickými členy lze najít v knize užitečné informace, například analytické odvození citlivostí při změně plochy prutového prvku.

[3] DUKKIPATI, R. V. *Spatial Mechanisms : Analysis and Systems*. Boca Raton : CRC Press, 2001. 363 s. ISBN 0849309328.

Text se zabývá analýzou a syntézou mechanismů s otevřenými i uzavřenými řetězci. V úvodu je krátce shrnuta klasifikace mechanismů včetně rozboru pohyblivosti mechanismů. Zde chybí popis práce s mechanismy, které se skládají z více kinematických smyček.

Dále je popsáno osm metod analýzy mechanismů. Kromě Hartenberg-Deavitovy metody, která je využívána v oblasti robotiky při analýze otevřených řetězců, a metody transformačních matic jsou ostatní prezentované metody těžce využitelné v technické praxi, protože vyžadují komplikované úvahy a mají značné nároky na schopnosti řešitele. Například metoda

generovaných ploch je aplikovatelná pouze na mechanismy, kde je přítomen alespoň jeden sférický kloub, a je založena na dvou prostorových ploch. Je zřejmé, že pro technickou praxi, kde je zapotřebí metod obecných, jednoduchých a automatizovatelných, nelze tuto metodu použít. Metoda transformačních matic je v díle popsána méně srozumitelně a velice stručně.

Pasáž zabývající se syntézou mechanismů popisuje pouze kolokační metodu. Chybí implementace optimalizačních algoritmů.

[16] CHARVÁT, J. *Syntéza mechanismů : Část 1*. Liberec : Vysoká škola strojní a textilní v Liberci, 1966. 154 s.

Kniha popisuje pouze grafické metody syntézy, které jsou dnes již nepoužívané z důvodu vysokého stupně vývoje výpočetní techniky.

[4] MALLIK, A. K.; GHOSH, A.; DITTRICH, G. K. *Kinematic Analysis and Synthesis of Mechanisms*. Boca Raton : CRC Press, 1994. 688 s. ISBN 9780849391217.

Jako metodu analýzy mechanismů uvádí kniha grafickou metodu, vektorovou metodu a metodu transformačních matic, která je však popsána velice stručně.

Pro syntézu mechanismů jsou navrženy grafické a analytické metody. Použití optimalizačních algoritmů je v knize pouze krátce zmíněno.

[48] SLAVÍK, J.; STEJSKAL, V.; ZEMAN, V. *Základy dynamiky strojů*. Praha : Vydavatelství ČVUT, 1997. 319 s. ISBN 80-01-01622-6.

Kniha popisuje teorii dynamiky mechanismů řešené maticovou transformační metodou, problematiku poddajných členů a vazeb mechanismů a také kmitání mechanismů. Dále je popsána optimalizace mechanismů z hlediska dynamických vlastností.

[11] BOCKO, J.; DELYOVÁ, I. *Optimalizácia mechanických sústav*Optimalizácia mechanických sústav. Košice : Technická univerzita v Košiciach, Strojnícka fakulta, 2010. 105 s. ISBN 978-80-553-0346-8.

Kniha se věnuje hlavně teoretickému popisu metod matematické optimalizace, a to analytickými i numerickými metodami. Chybí popis některých moderních metod jako je například metoda Broyden-Fletcher-Goldfarb-Shanno, metody typu „Trust Region“ nebo metody jednorozměrného hledání využívající Armijo a Goldteisnovo pravidlo. Cenná je kapitola zabývající se genetickým algoritmem, jehož použití je demonstrováno na příkladu.

[12] JENSEN, O. F.; HANSEN, J. M. Dimensional Synthesis of Spatial Mechanisms and the Problem of Non-Assembly. *Multibody System Dynamics*. 2006, 15, 2, s. 107-133. ISSN

1384-5640.

Článek popisuje komplexní proceduru syntézy vodících mechanismů. Zvláštní pozornost je věnována problematice nazývané v článku „Non-Assembly“. Při optimalizačním procesu totiž algoritmus vyčísľuje cílovou funkci pro různé rozměrové konfigurace mechanismu. Pokud optimalizační algoritmus navrhne rozměry, které neumožňují mechanismus sestavit, potom úloha polohy nekonverguje, není tudíž možné vyčísľit hodnotu cílové funkce a optimalizační algoritmus havaruje. Autoři článku řeší problém následovně. Zdvihová závislost se skládá ze soustavy nelineárních rovnic. V případě, že mechanismus lze sestavit a je nalezeno řešení soustavy, se všechny tyto rovnice rovnají nule. V opačném případě jsou rezidua těchto rovnic nenulová a soustava nekonverguje. Zde autoři článku navrhuji použít optimalizaci a minimalizovat rezidua, poté vyčísľit hodnotu cílové funkce a optimalizační proces může pokračovat.

Dále zkoumána problematika nejednoznačnosti řešení zdvihové závislosti. Jinými slovy pro jednu hodnotu vstupní souřadnice může existovat několik variant poloh mechanismu. Tuto část článku nepovažuje autor za podstatnou, protože tuto překážku lze ve většině překonat pomocí počátečních odhadů polohy a dostatečně malého kroku vstupních souřadnic.

[49] LEDERER, H.; LONI, G.; GORVES, B. Application of higher order derivatives in the synthesis of crank and cam mechanisms. *Computational Kinematics* [online]. 2009, ix, [cit. 2011-01-13]. Dostupný z WWW: <<http://www.springerlink.com/content/gqr810g4m38lv10q/fulltext.pdf>>.

Odborný text popisuje využití derivací vyššího řádu při návrhu klikových a vačkových mechanismů. Veškerý matematický popis je proveden v komplexní rovině a tudíž je vhodný pro řešení rovinných mechanismů. Využitelnost u vodících mechanismů je možná v případě, že bychom chtěli dosáhnout toho, aby se určitý bod mechanismu pohyboval například po přímce nebo po trajektorii s danou křivostí.

[50] MARTÍNEZ-ALFARO, H. Four-bar Mechanism Synthesis for n Desired Path Points Using Simulated Annealing. In *Advances in Metaheuristics for Hard Optimization* [online]. Berlin : Springer, 2007 [cit. 2011-01-13]. Dostupné z WWW: <<http://www.springerlink.com/content/gqr810g4m38lv10q/fulltext.pdf>>. ISBN 978-3-540-72960-0.

Autor popisuje využití „simulated annealing algorithm“ k syntéze vodícího mechanismu tak, aby jistý bod soustavy co nejlépe opisoval trajektorii danou n body. Metoda je popsána na jednoduchém čtyřkloubovém mechanismu. Algoritmus syntézy je založen na náhodné změně rozměrů mechanismu a následovném vyhodnocení za pomoci pravděpodobnosti.



[13] HANSEN, J. M. Synthesis of Mechanisms. In *Advanced Design of Mechanical Systems: From Analysis to Optimization* [online]. Vienna : Springer, 2009 [cit. 2011-01-13]. Dostupné z WWW: <<http://www.springerlink.com/content/k3uwl6q5488768lk/fulltext.pdf>>. ISBN 978-3-211-99461-0.

Text popisuje tři způsoby syntézy mechanismů. První způsob je založen na časově proměnných rozměrech mechanismu, kdy vodící mechanismus exaktně opisuje potřebnou dráhu a aby bylo toto splněno, rozměry mechanismu jsou měněny. Následovně pomocí optimalizačního procesu je vybrána ta varianta mechanismu, u které se během pohybu mění rozměry mechanismu nejméně. Druhá navržená a jak autoři uvádí málo prozkoumaná metoda nahrazuje některé členy mechanismu pružinou s tlumičem. Poslední metodou syntézy je optimalizační metoda, která umožňuje stavy „Non-Assembly“, které rozměrově neumožňují mechanismus sestavit.

[51] KIM, Y. Y., et al Configuration Design of Rigid Link Mechanisms by an Optimization Method: A First Step. In *IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials* [online]. Netherlands : Springer, 2006 [cit. 2011-01-13]. Dostupné z WWW: <<http://www.springerlink.com/content/1211um530888j12x/fulltext.pdf>>. ISBN 978-1-4020-4752-7.

Text popisuje metodu automatizované topologické syntézy rovinného mechanismu, který je složen z kloubově spojených prutů. Idea metody spočívá v tom, že rovinný prostor je rozdělen na bloky, které jsou v rozích vzájemně spojeny pružinami. Předepsaným způsobem s bloky pohybuje a porovnáváme hodnoty tuhostí, které jsou nastavovány pomocí optimalizační metody. Dle srovnání tuhostí pružin jednotlivé bloky spojujeme kloubovou nebo pevnou vazbou, popřípadě rozpojíme. Tímto iterativním postupem získáme optimální topologii.

[52] RAI, A. K.; SAXENA, A.; MANKAME, N. D. Unified synthesis of compact planar path-generating linkages with rigid and deformable members. *Structural and Multidisciplinary Optimization* [online]. 2002, 6, [cit. 2011-01-13]. Dostupný z WWW: <<http://www.springerlink.com/content/n696j0w8752477h6/fulltext.pdf>>. ISSN 1615-1488.

Autoři popisují syntézu mechanismů složených z poddajných prutů. Jsou popsány tuhosti členů. Syntéza je provedena pomocí optimalizace.

[53] STOLPE, M.; ACHTZIGER, W.; KAWAMOTO, A. A Concept for Global Optimization of Topology Design Problems. In *IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials* [online]. Netherlands : Springer, 2006 [cit. 2011-01-13]. Dostupné z WWW:

<<http://www.springerlink.com/content/1855x46768527317/fulltext.pdf>>. ISBN 978-1-4020-4752-7.

Text popisuje návrh optimální topologie rovinných prutových konstrukcí a rovinných mechanismů složených z prutů. Topologie mechanismu je navrhována následujícím postupem. Nejdříve je vytvořena síť klubů, které jsou spojeny elastickými pruty všemi možnými způsoby. Následovně takto vzniklou prutovou soustavu zatížíme a porovnáme deformace jednotlivých prutů. Nejvíce deformované pruty odebereme a iterativním postupem se blížíme k optimální topologii.

[54] EBERHARD, P.; SEDLACZEK, K. Grid-Based Topology Optimization of Rigid Body Mechanisms. In *Advanced Design of Mechanical Systems: From Analysis to Optimization* [online]. Vienna : Springer, 2009 [cit. 2011-01-13]. Dostupné z WWW: <<http://www.springerlink.com/content/t18tu017v8631214/fulltext.pdf>>. ISBN 978-3-211-99461-0.

Článek se zabývá topologickou syntézou rovinných mechanismů složených z prutů. Princip návrhu topologie je následující. Nejdříve je vytvořena síť bodů neboli kloubů, které jsou propojeny tuhými pruty všemi možnými způsoby. Následuje minimalizace reziduí rovnic úlohy polohy a rovnic popisujících vstupní a výstupní pohyb mechanismu. Minimalizací vybereme sydu prutů tak, abychom získali optimální trajektorii.

### 11.16.2 Matematická teorie optimalizace a řešení soustav nelineárních, algebraických rovnic

[15] NOCEDAL, J.; WRIGHT, S. J. *Numerical Optimization*. New York : Springer, 2000. 636 s. ISBN 0-387-98793-2.

Dílo je zaměřeno na podrobný a názorný výklad principů většiny moderních matematických metod optimalizace, a to včetně jejich odvození a veškerých důkazů. Uvedeny jsou také algoritmy, ale těmto není věnována hlavní pozornost. Kniha obsahuje také rozbor principů metod řešení soustav nelineárních rovnic, což je nutné při řešení úlohy polohy mechanismů. Chybí pouze rozbor genetických algoritmů. Pro pochopení principu a základních myšlenek optimalizačních metod je kniha nepostradatelná.

[17] SUN, W.; YUAN, Y. *Optimization Theory and Methods Optimization Theory and Methods : Nonlinear Programming*. New York : Springer, 2006. 687 s. ISBN 978-0-387-24975-9.

V knize je popsána teorie metod optimalizace včetně algoritmů těchto metod. V díle chybí rozbor genetických algoritmů. Teoretické pasáže jsou méně srozumitelné. Naopak algoritmy jsou popsány do nejmenších detailů a jsou funkční, avšak nejsou uvedena srovnání a numerické testy jednotlivých algoritmů.

[22] YANG, W. Y., et al. *Applied Numerical Methods Using MATLAB*. New Jersey : Wiley, 2005. 528 s. ISBN 978-0-471-69833-3.

Dílo se zabývá implementací numerických metod a jejich algoritmů do programového prostředí MATLAB. Využitelné jsou zvláště kapitoly zabývající se optimalizací a řešením soustav nelineárních a diferenciálních rovnic. Popis řešení soustav diferenciálních rovnic lze využít při uvažování problémů z oblasti dynamiky mechanismů.

[45] ZELINKA, I. *Umělá inteligence v problémech globální optimalizace*. Praha : BEN – technická literatura, 2002. 192 s. ISBN 80-7300-069-5.

Kniha popisuje dva nové algoritmy evolučního charakteru, a to samo-organizující se migrační algoritmus a algoritmus diferenciální evoluce. Popsané metody jsou použity na příkladech z technické praxe. Oba algoritmy lze popsat jako netradiční jejich efektivitu při syntéze mechanismů by bylo užitečné zjistit.

[18] FLETCHER, R. *Practical Methods of Optimization*. Cornwall : Wiley, 2000. 450 s. ISBN 978-0-471-49463-8.

Kniha se věnuje vysvětlení základních principů optimalizačních metod, jejich odvozením a důkazům. Velmi přínosné jsou numerické testy metod a vzájemné srovnání jejich efektivit. Autor knihy je jeden ze zakladatelů často používané metody BFGS. Tato metoda je v díle detailně a srozumitelně popsána a srovnána s ostatními metodami optimalizace bez omezení.

[19] RARDIN, R. L. *Optimization in Operations Research*. New Jersey : Prentice Hall, 2000. 919 s. ISBN 0-02-398415-5.

Kniha je psána ve formě přehledné příručky s cílem, aby čtenář byl schopen okamžitě implementovat metodu na svoji problematiku. Nelineárními problémy se zabývají poslední dvě kapitoly knihy. Tyto které jsou tedy použitelné pro optimalizaci mechanismů. Kniha obsahuje mnoho příkladů použití optimalizace na praktické problémy.